# Composition of Interval and Probabilistic Timing Models

**Rolf Ernst**

**Matthias Ivers**

**TU Braunschweig**

# Overview

- motivation
- probabilistic models in systems integration
- scheduling analysis using probabilistic bounds
- combining WC intervals and probabilistic bounds
- conclusion

# Motivation 1/3

- **performance analysis and optimization based on best case/ worst case timing intervals have been established**
  - **all behavior captured, results not dependent on system state**
  - **conservative guarantees**
  - **compositionality (SymTA/S, MPA)**
  - **system predictability, sensitivity analysis possible**

# Motivation 2/3

- **combination with less conservative analysis requested**
  - **worst case is observed to occur very infrequently**
    - **dropping tasks or jobs is regular overload action**
    - **typical argument from practice**
  - **probabilistic QoS requirements**
    - **e.g. 99.9% correct frame sequence**
    - ***conservative result required***
  - **inclusion of statistic sources, such as transient faults**
    - **fault handling as exception or in regular WCET**
    - ***conservative results e.g. for safety analysis***
- **try probabilistic performance models and related analysis**
  - ***this talk***

# Requirements to Probabilistic Models and Analysis

- compositionality
- compatibility to WC models
  - needed for performance guarantees
  - needed for mixed criticality
    - e.g. safety critical with WC models, others probabilistic
- availability in practical design
  - derived from traces
  - derived from models

# Limitiations of Stochastic Analysis

- **typically steady state assumption**
- **execution time distribution is context dependent**
  - **composition may change execution time distribution**
  - **de Alfaro, Henzinger, Jhala (*Compositional Methods for Probabilistic Systems, 2001*):**
    - **composition of probabilistic and non-deterministic component models (such as interval based timing models) is subject to strong limitations - non-determinism must not influence output of component**
    - **for composition, any dependencies of transitions (on state) and actions must be declared and modeled explicitely**
      - **usually not available in practice**

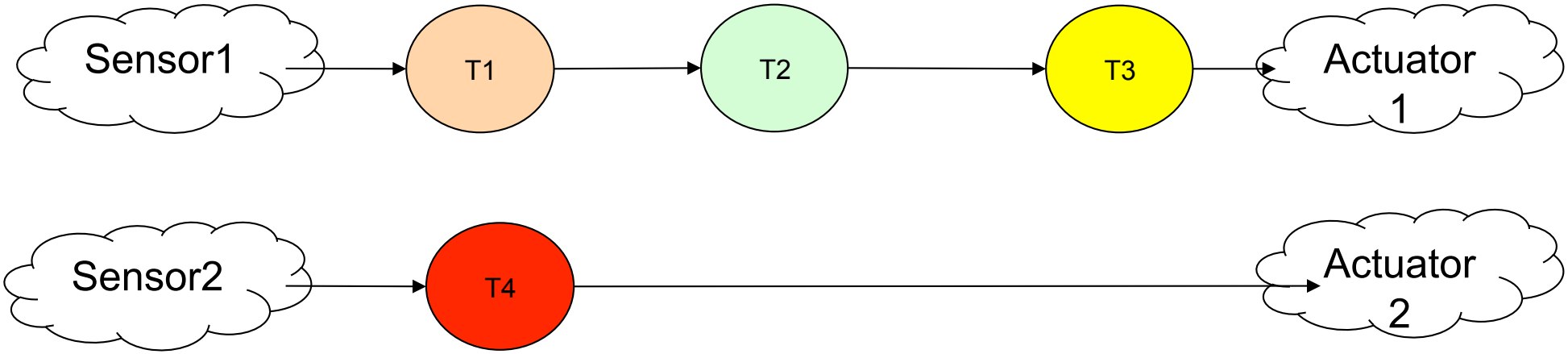
# Initial Requirements/Assumptions

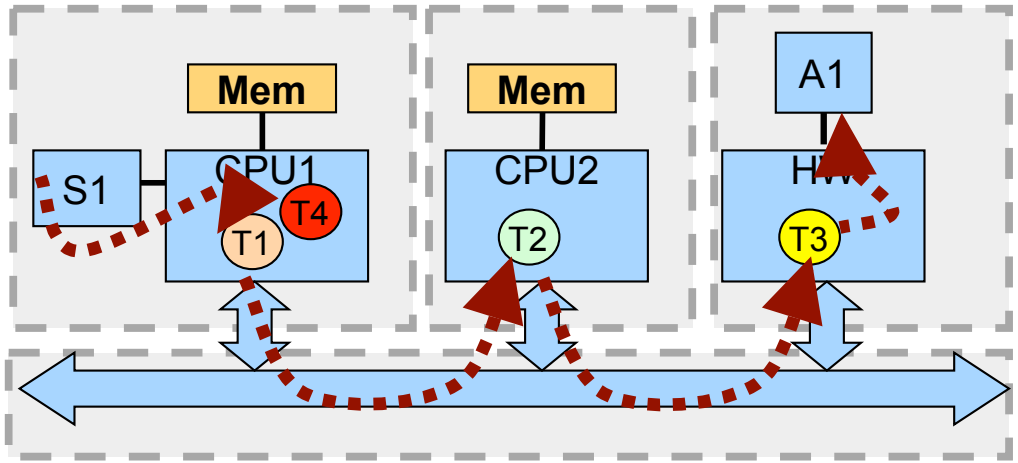
- effects of changed execution time distribution appear too complicated
  - require that task execution time distribution must remain unchanged in composition
  - reduce task timing to core execution time
  - treat CRPD and blocking as scheduling effects
- function execution (traces) remains unchanged in composition
  - our work focuses on modeling and analysis in systems integration
  - assume that core execution time does not change with integration (see above)

# Overview

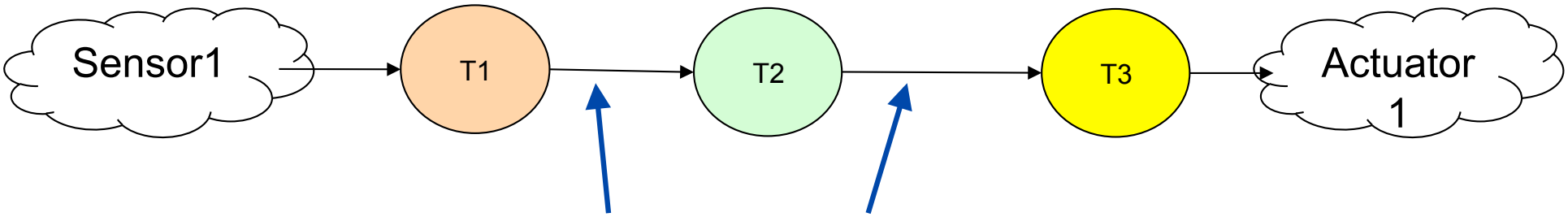
- motivation
- probabilistic models in systems integration
- scheduling analysis using probabilistic bounds
- combining WC intervals and probabilistic bounds
- conclusion

# Integration Example

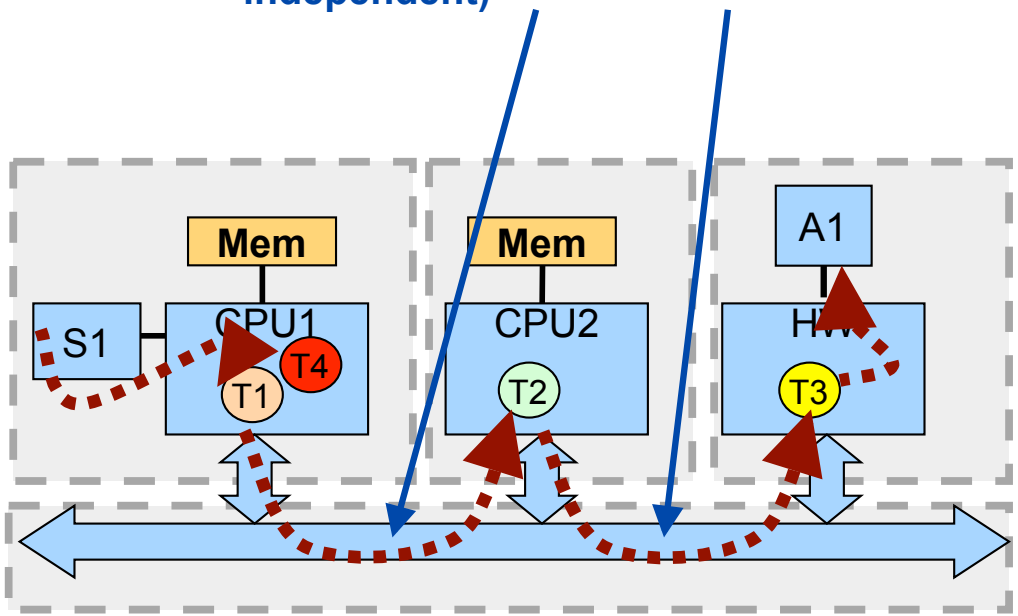


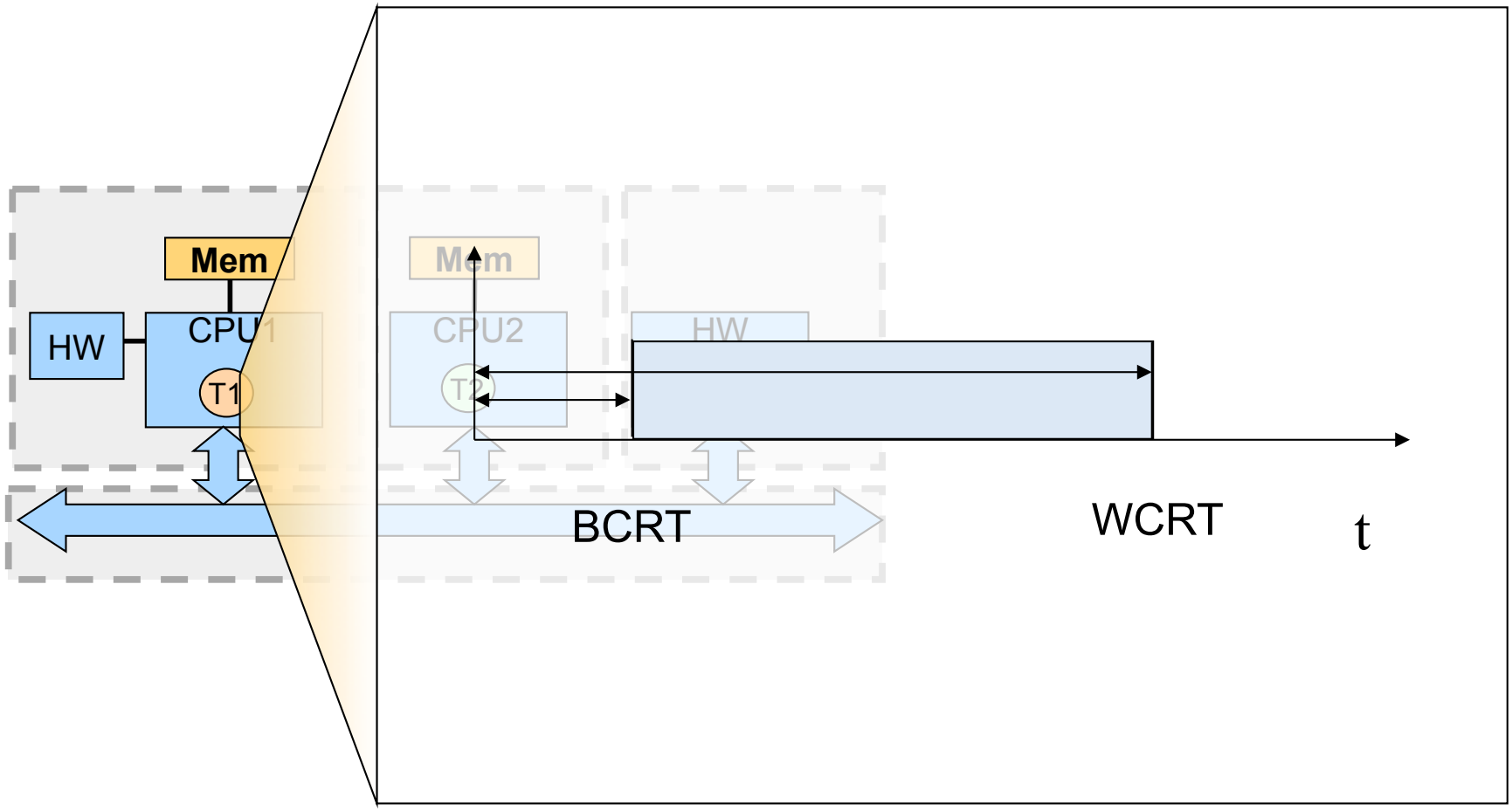

# Integration Example – Functional Dependency



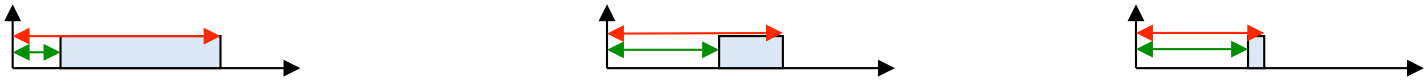

- **functional dependency of communicating tasks**
  - **dependency of communicating tasks cannot be avoided in realistic systems**
  - **resulting task execution times are dependent (if not enforced as state independent)**

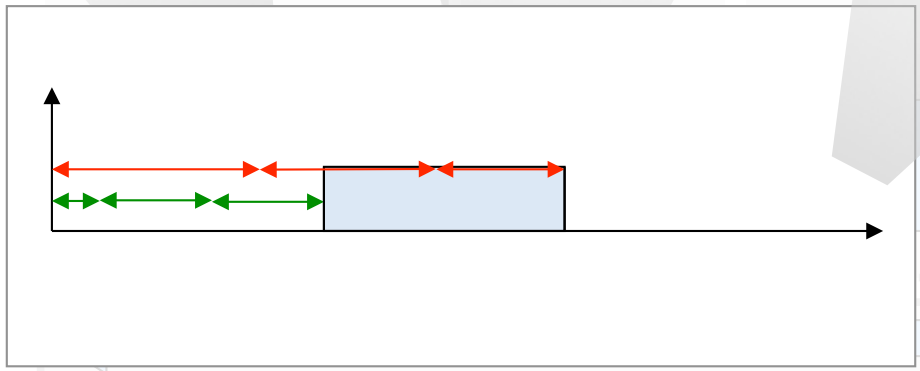



# Task Timing – WC Execution Time Interval

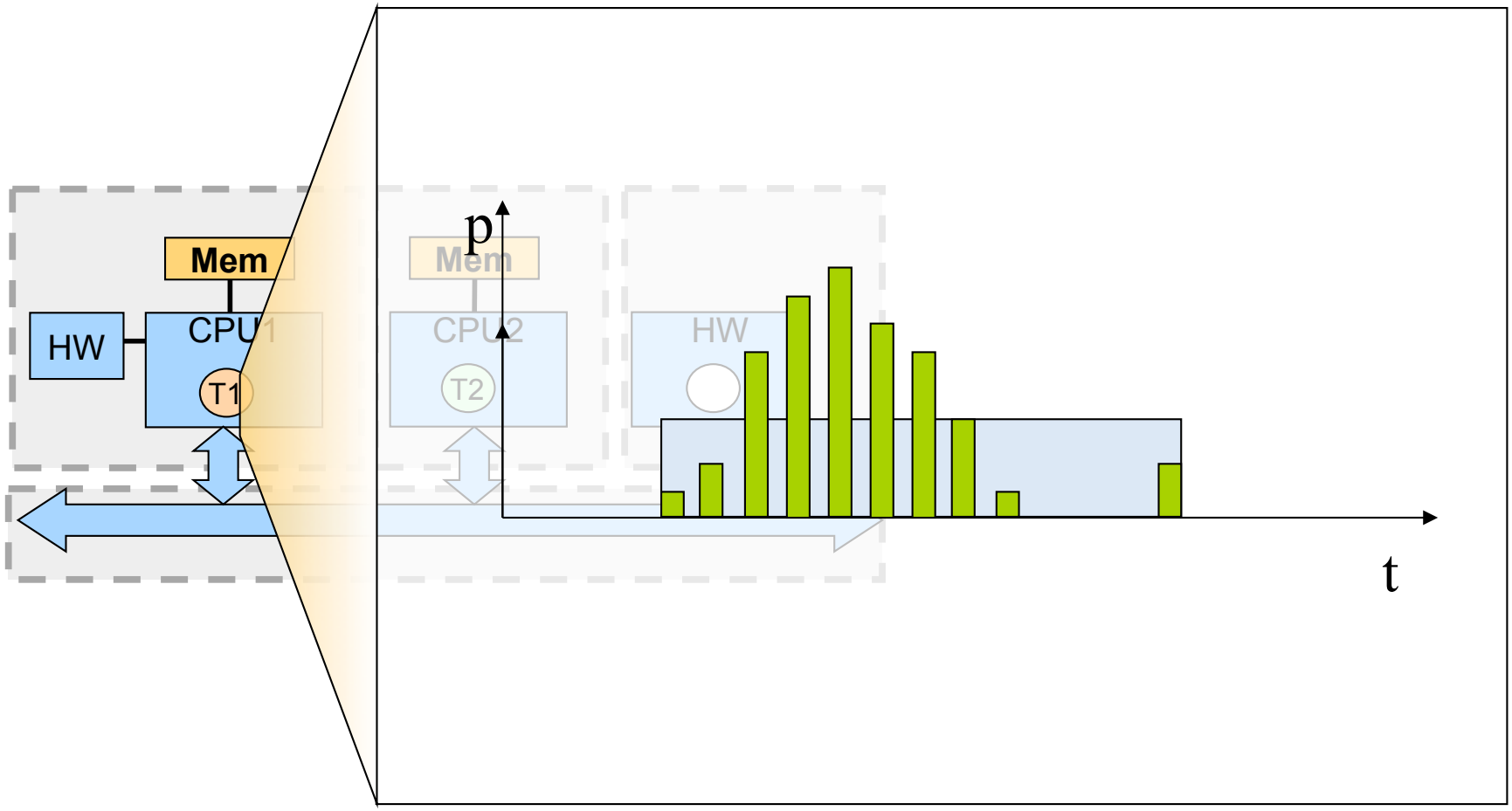


# End-to-End Response Time – WC Interval

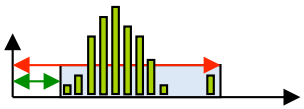


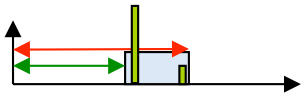


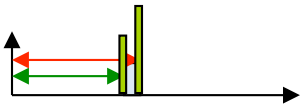
# Task Timing Distribution

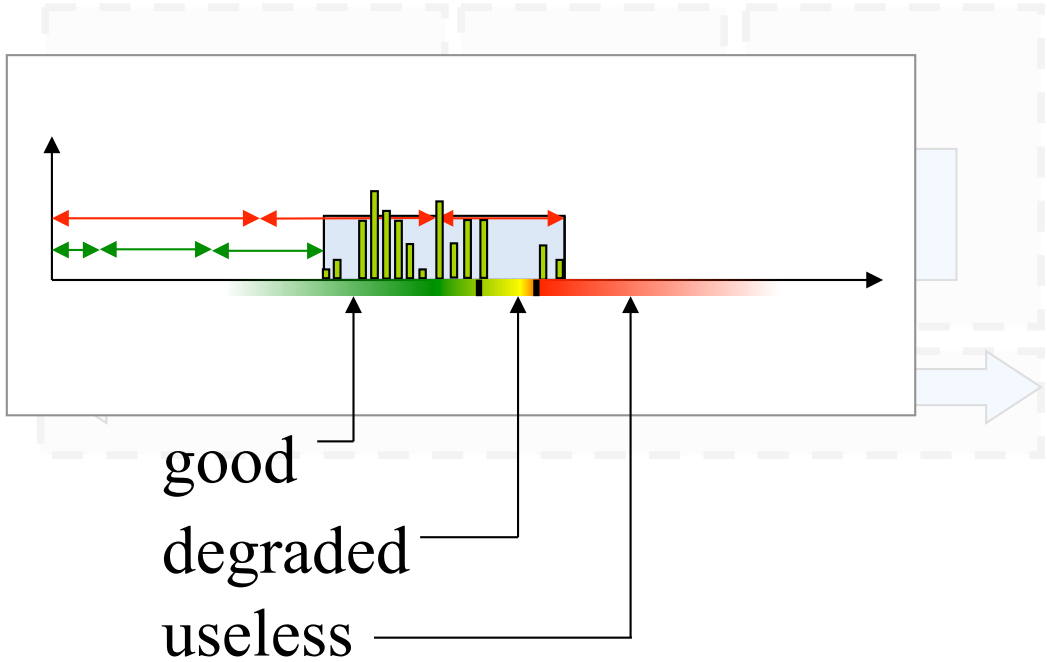


# Timing Distribution in Example

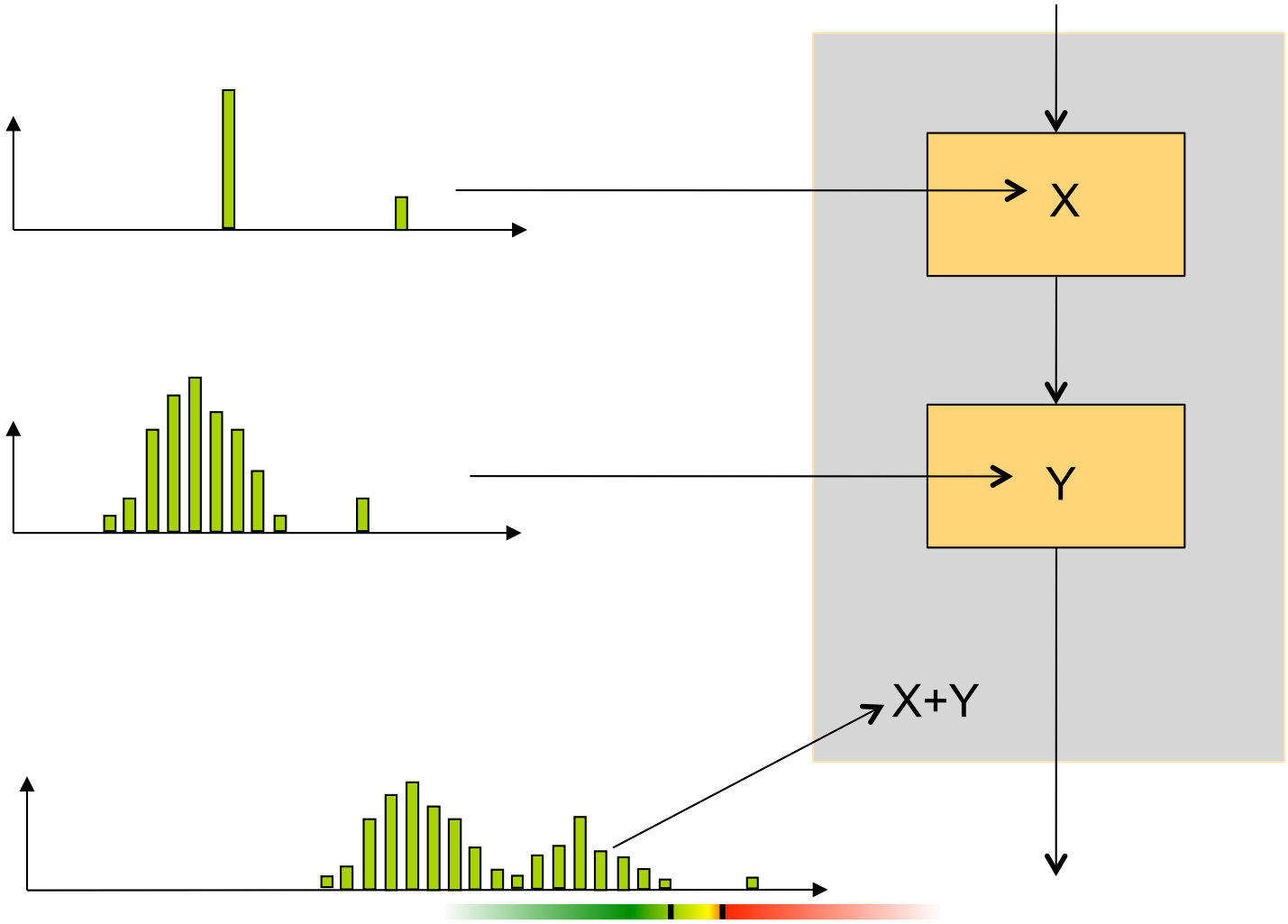




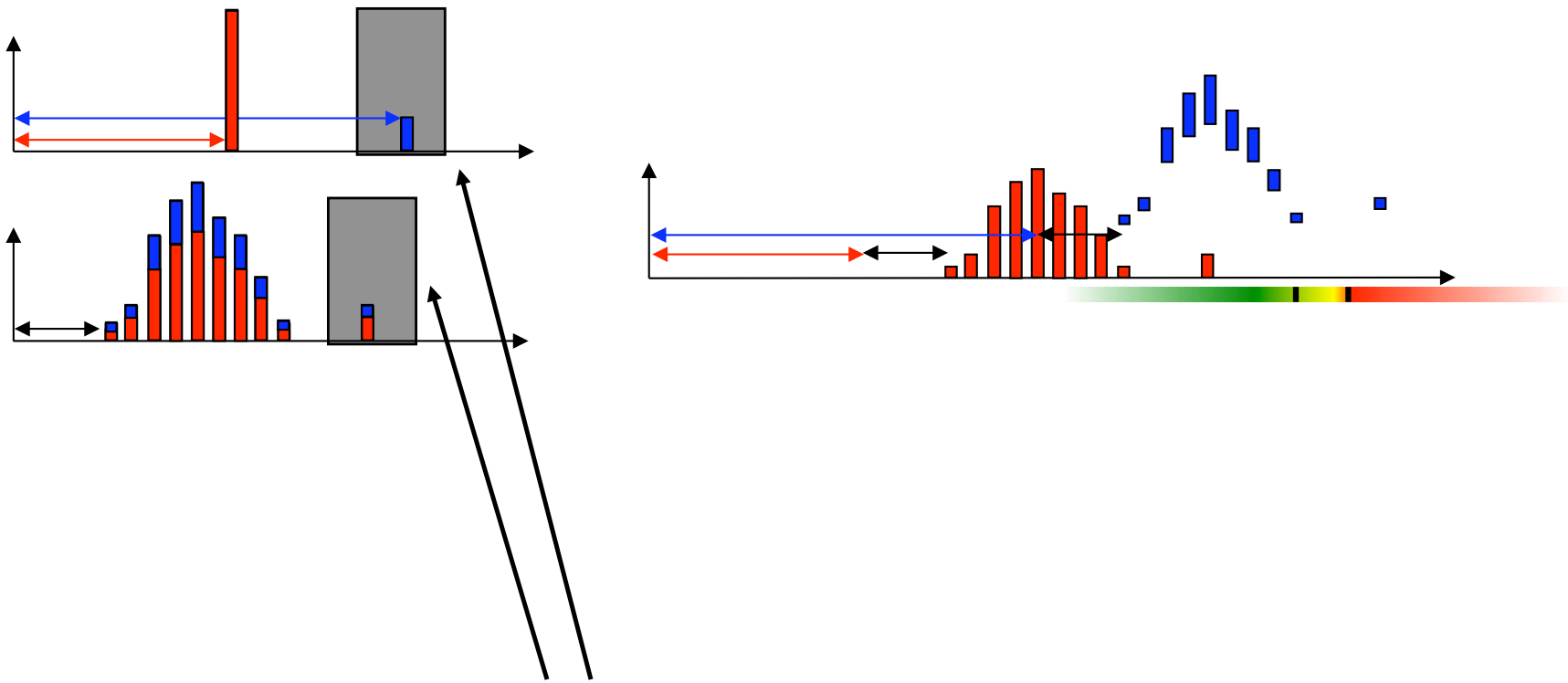




# Dependent Tasks

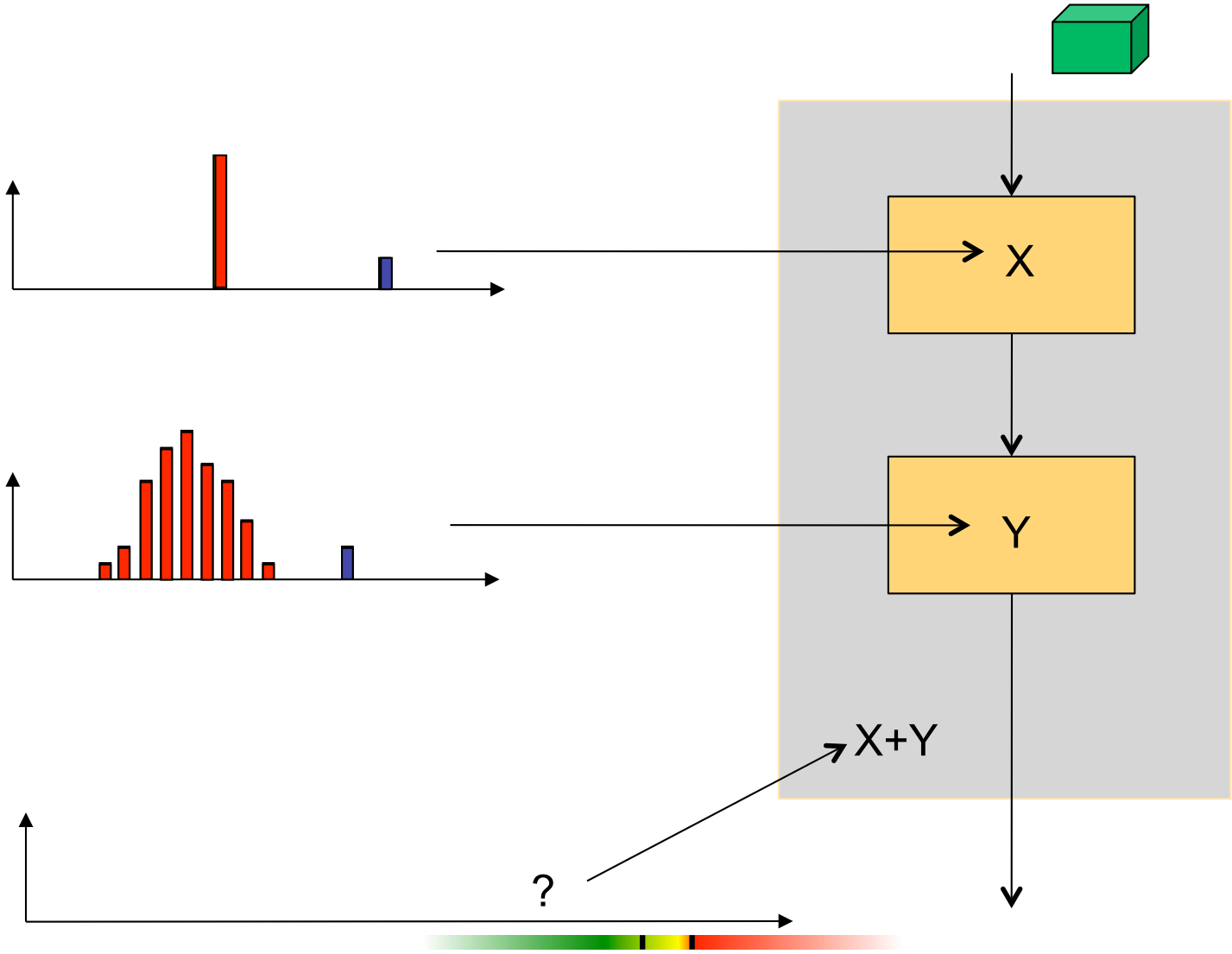


# Addition of two Random Variables

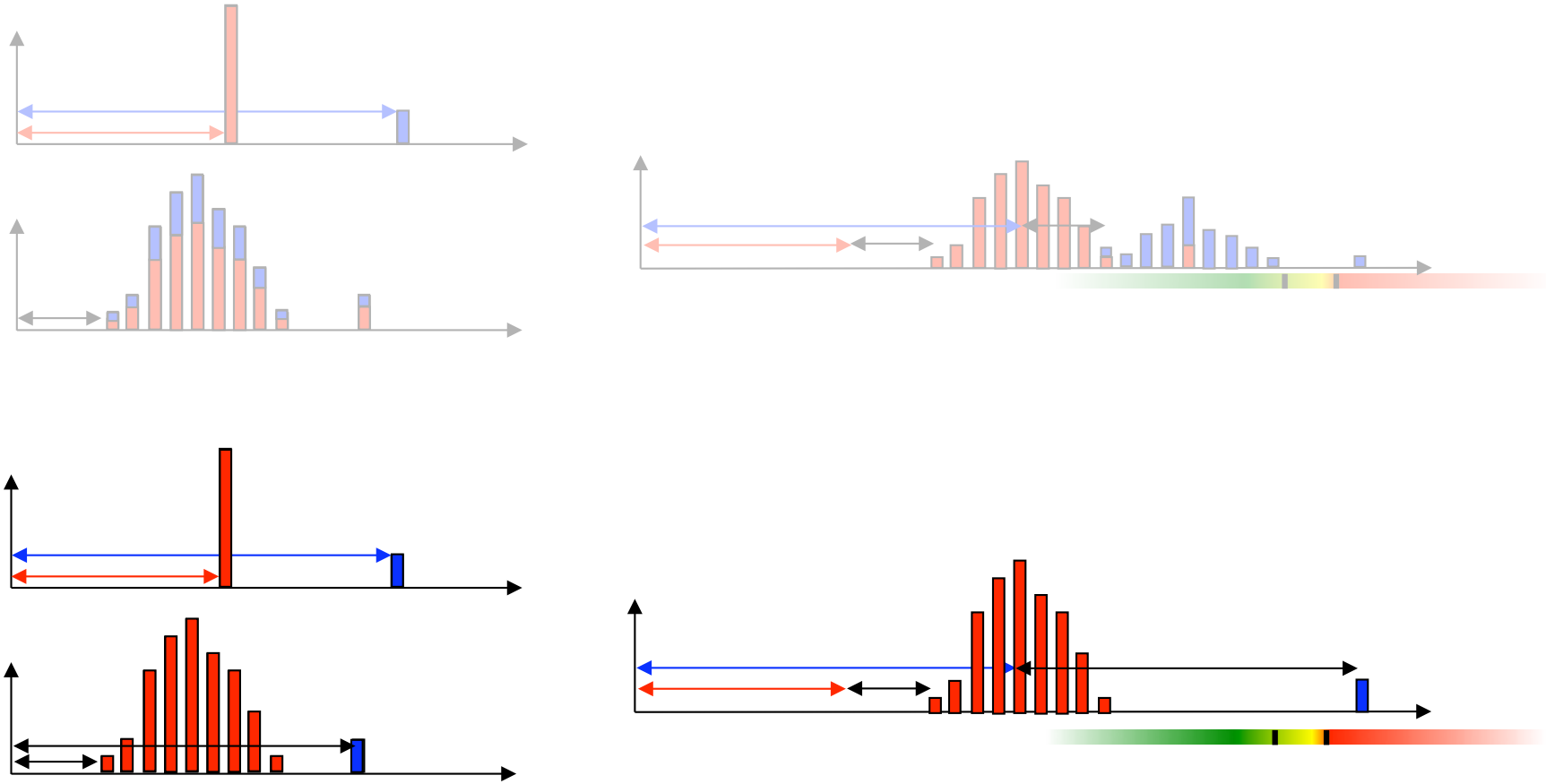


"I think these samples are from
error handling functions… in this case, they will always coincide!"

# Dependent Random Variables

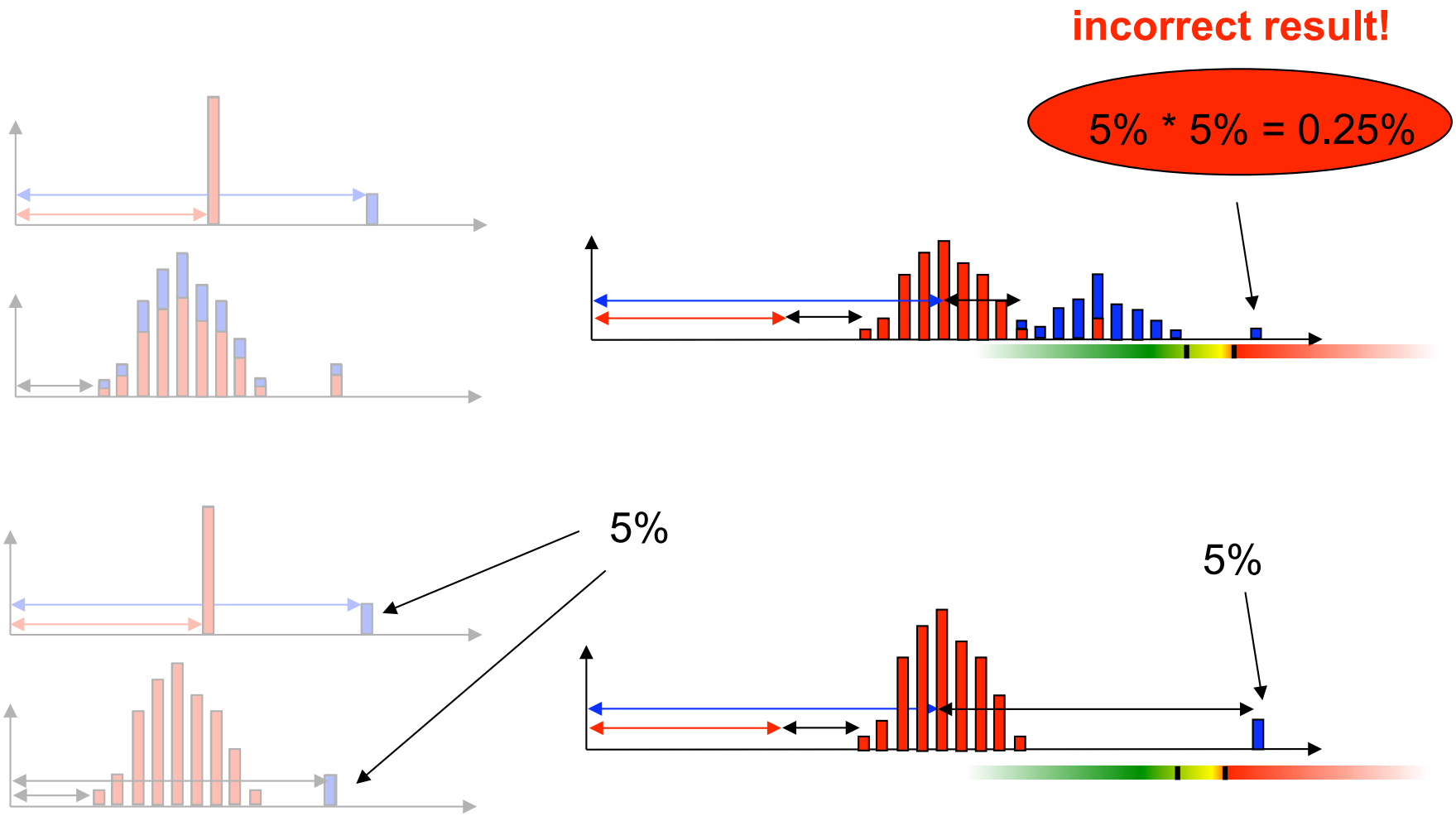


# Dependent Random Variables

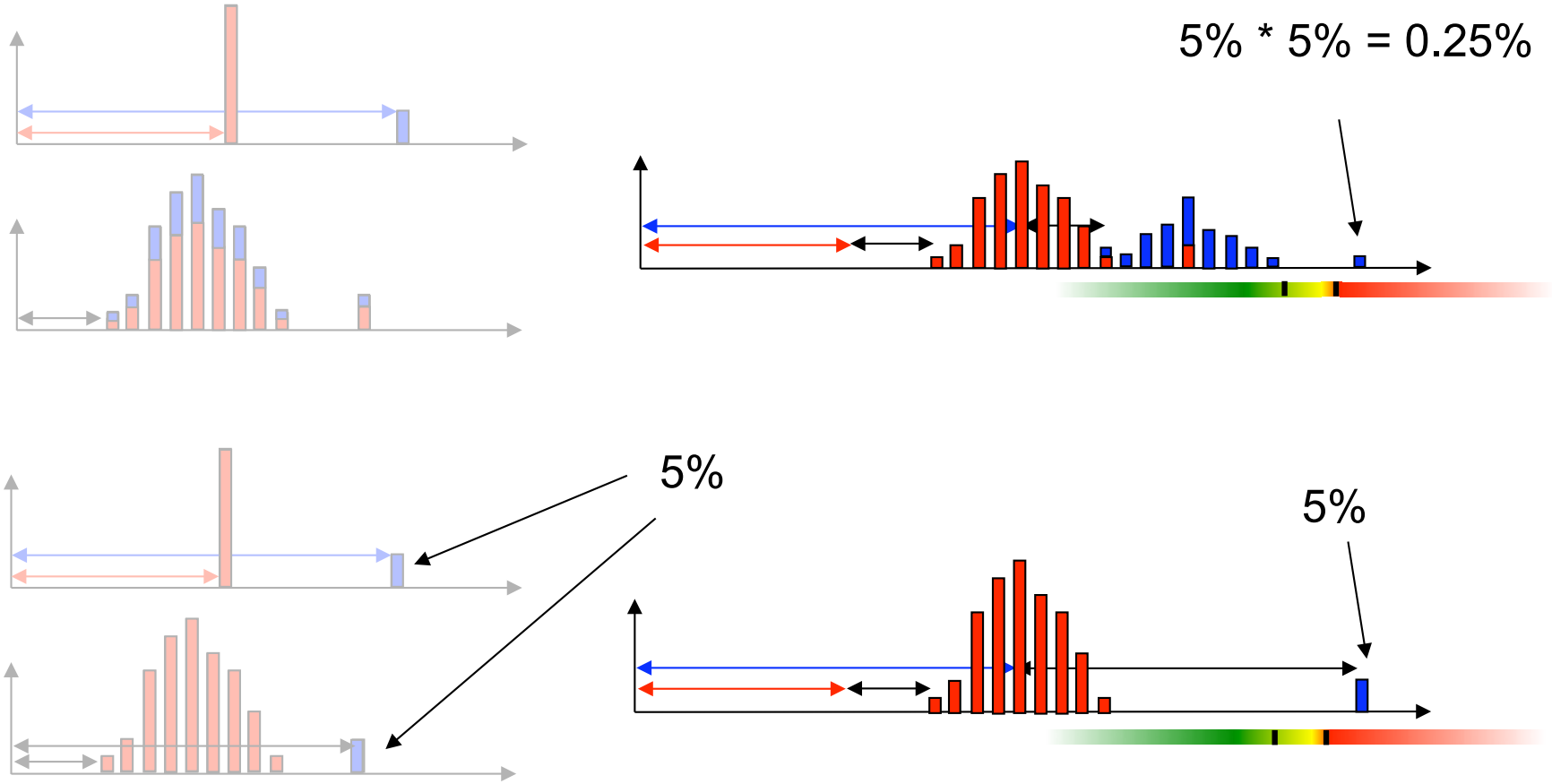


**„co-monotonous“ execution time distributions**

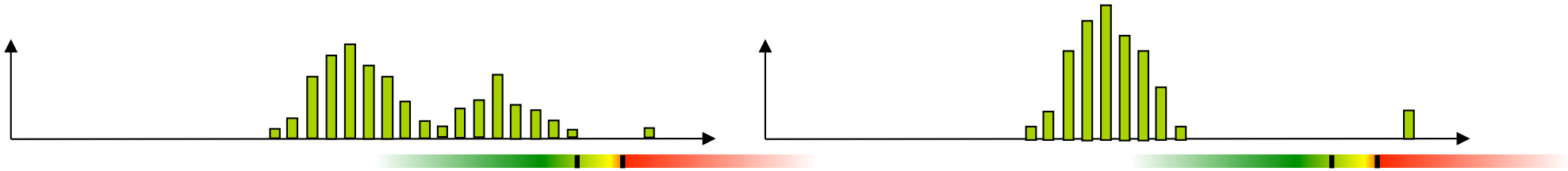
# Underestimated Error



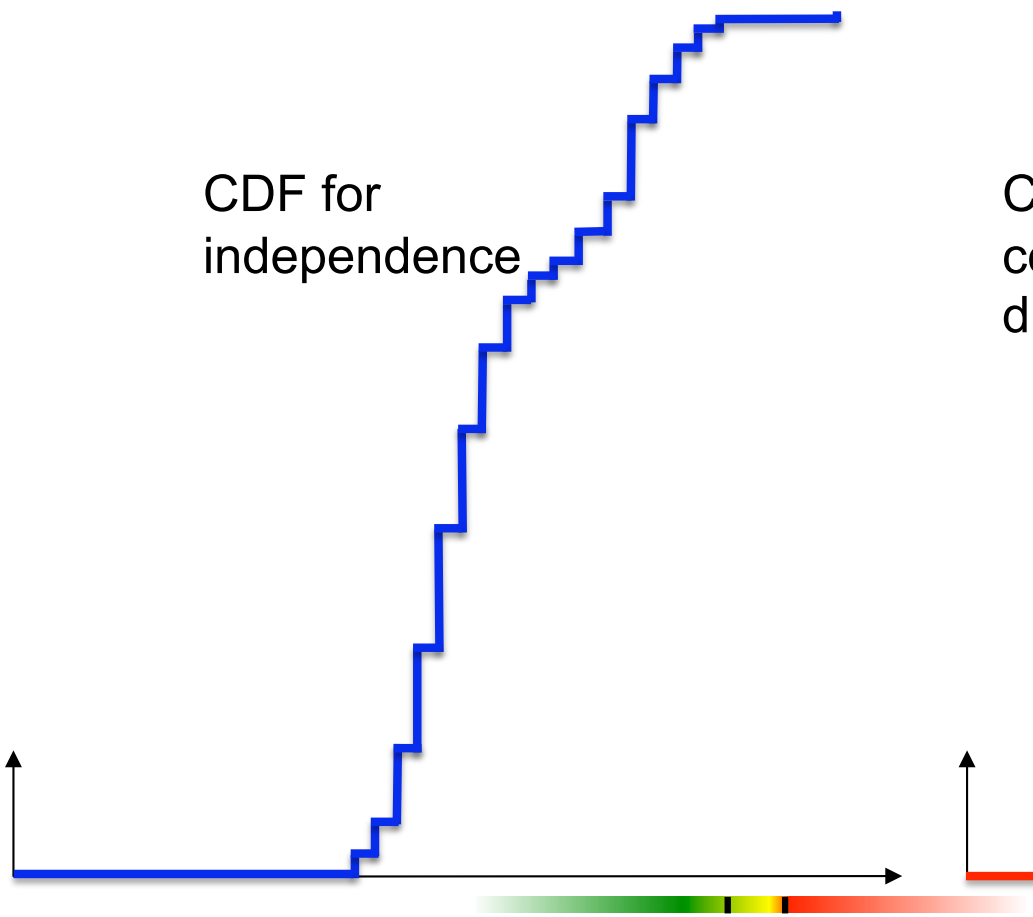
# Underestimated Error

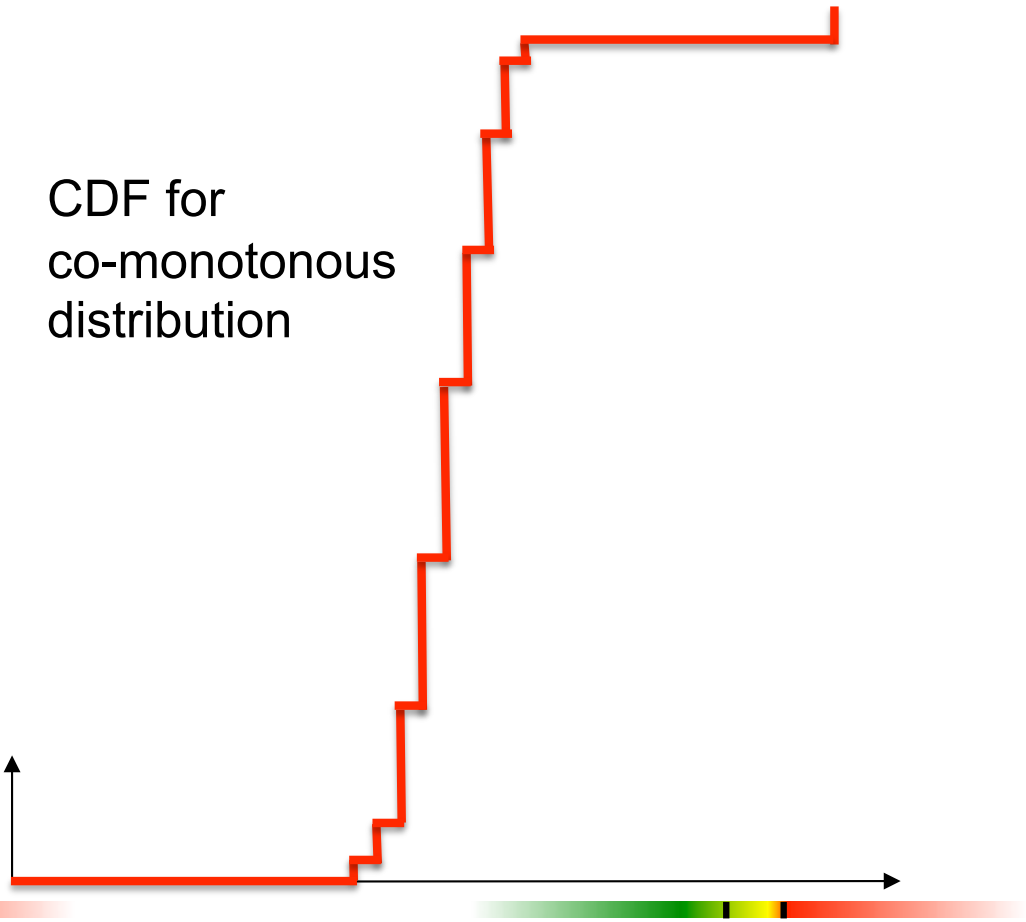


# Stochastic & Uncertainty

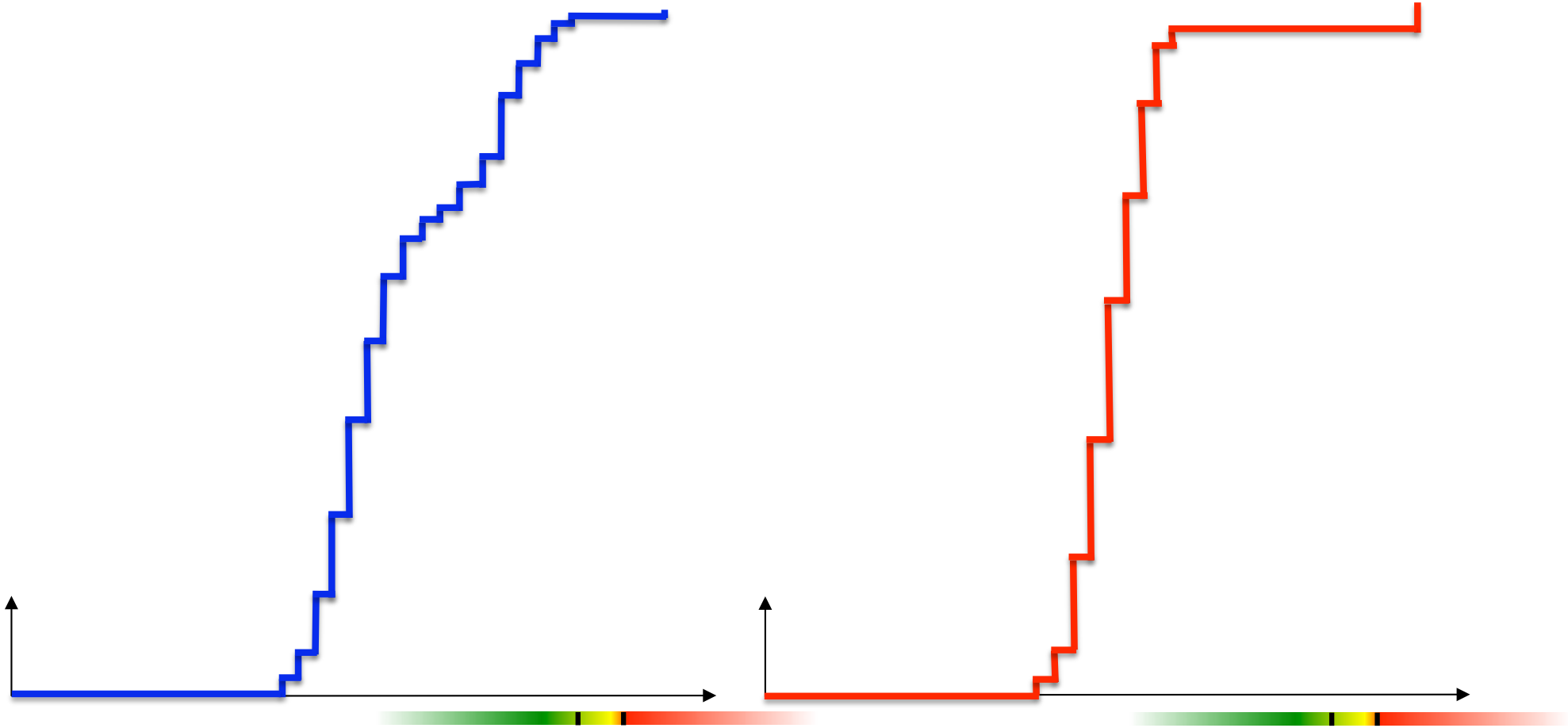


# Stochastic & Uncertainty

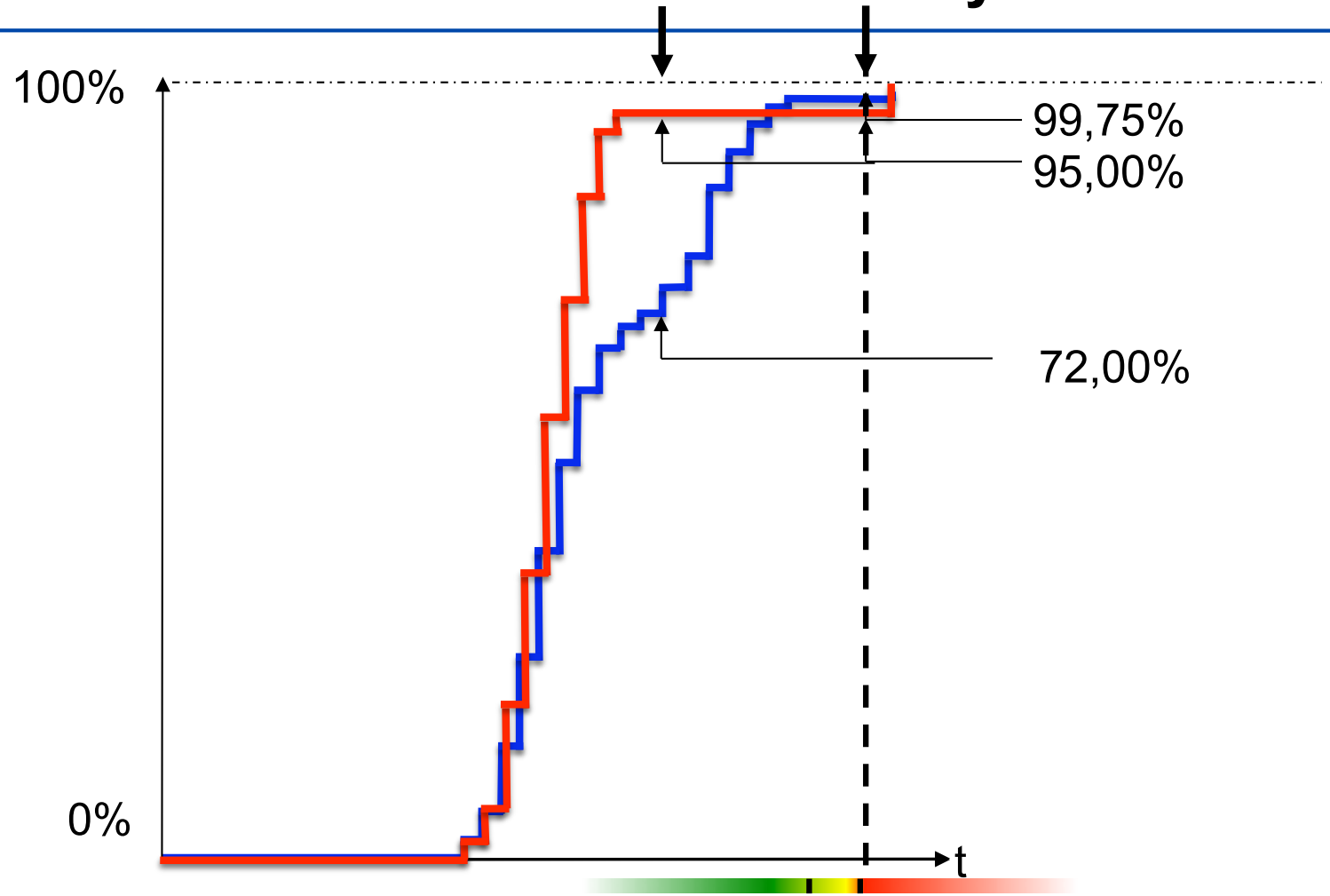


# Stochastic & Uncertainty



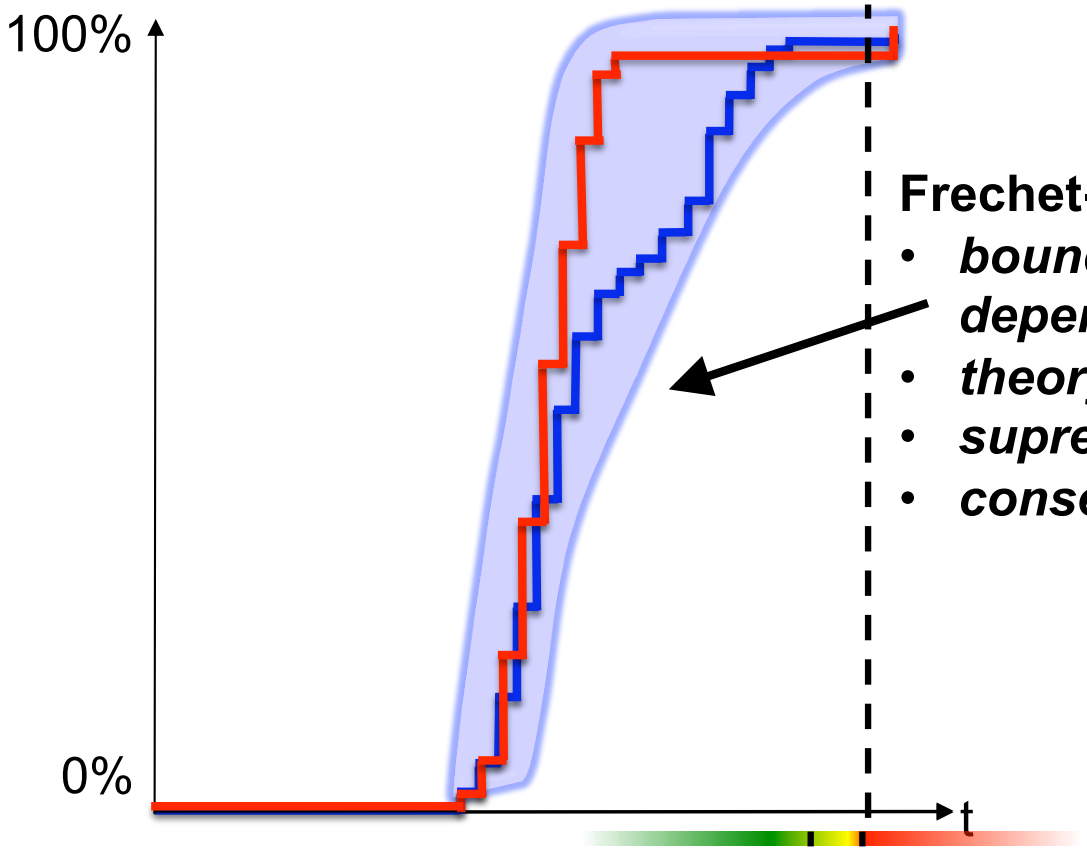
# Stochastic & Uncertainty




⇒ comonotonous distribution is not supremal

⇒ not conservative

# Stochastic & Uncertainty

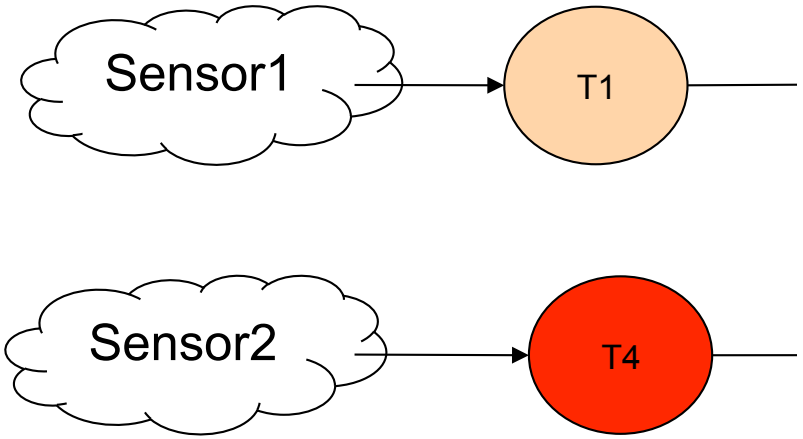



**Frechet-Höffding bounds**

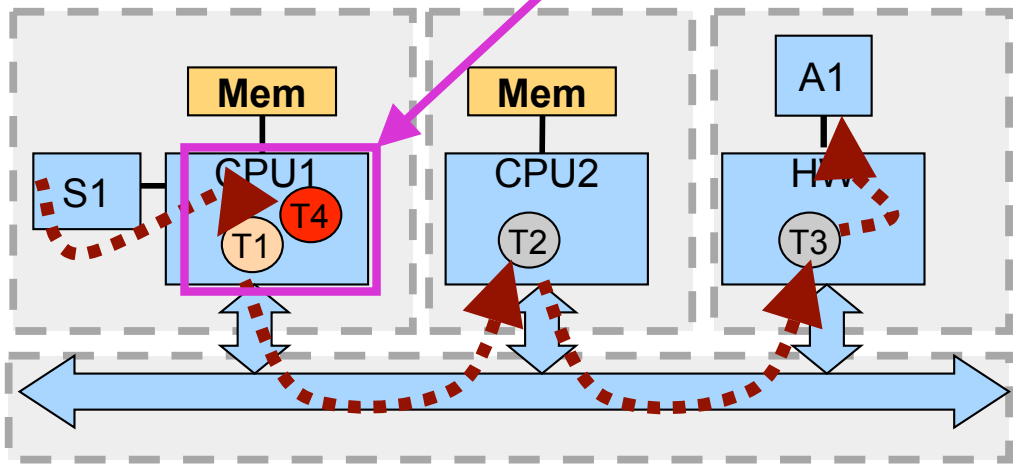
- ***bounds over all possible dependencies***
- ***theory available***
- ***supremal***
- ***conservative***

- **using Frechet bounds for stochastic WCET analysis proposed by Newly, Bernat, Burns 2005**
- **not yet used for schedulability and response time analysis**

# Integration Example – More Dependencies



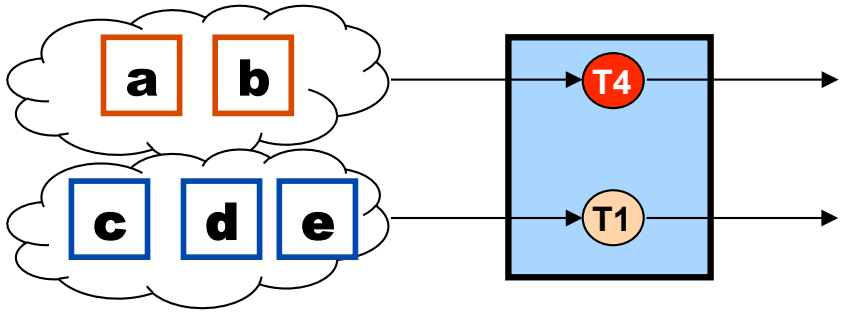

- **execution time dependency via shared ressources**
  - **task timing coupled via platform scheduling**
  - **additional side effects, such as CRPD**




# Analysis of Scheduled Stochastic Systems

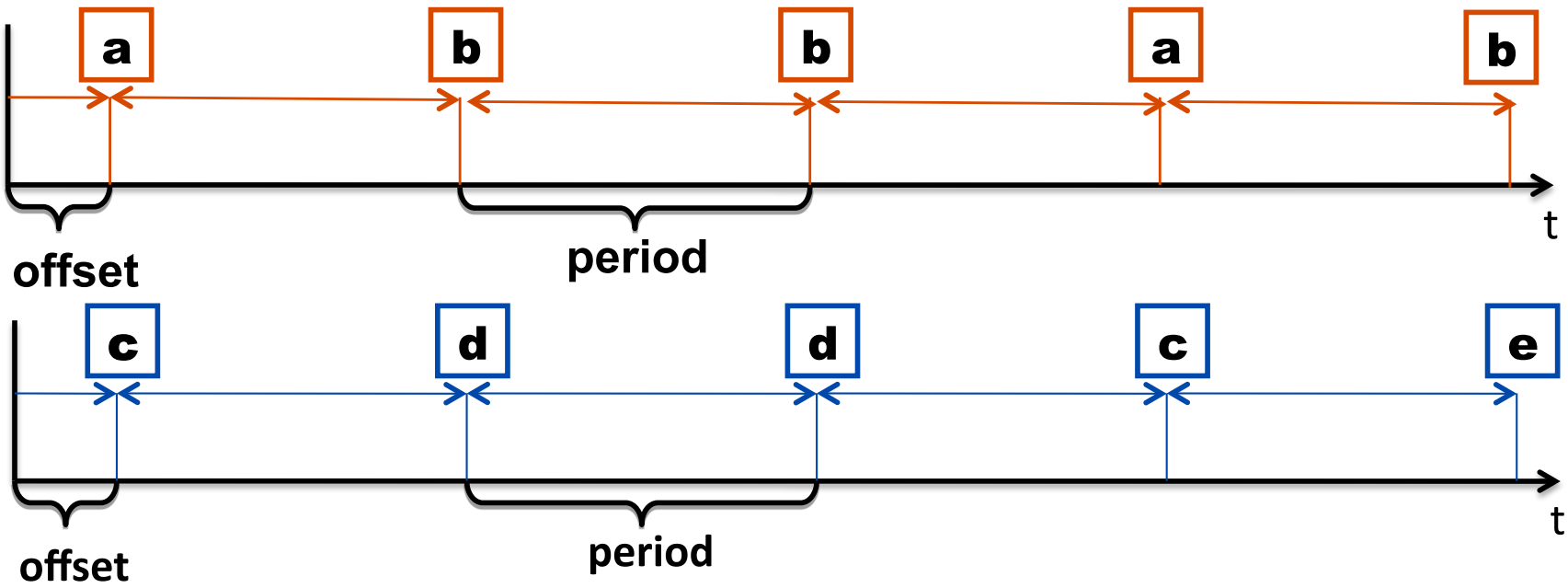
- **scheduling Analysis of Stochastic Task Models**
  - **Gardner et al. 1999 STDA**
  - **Diaz et al 2005**
  - **Manolache et al 2005**
- **assumptions**
  - **task independence**
  - **job independence**

# Integration Example – Scheduling

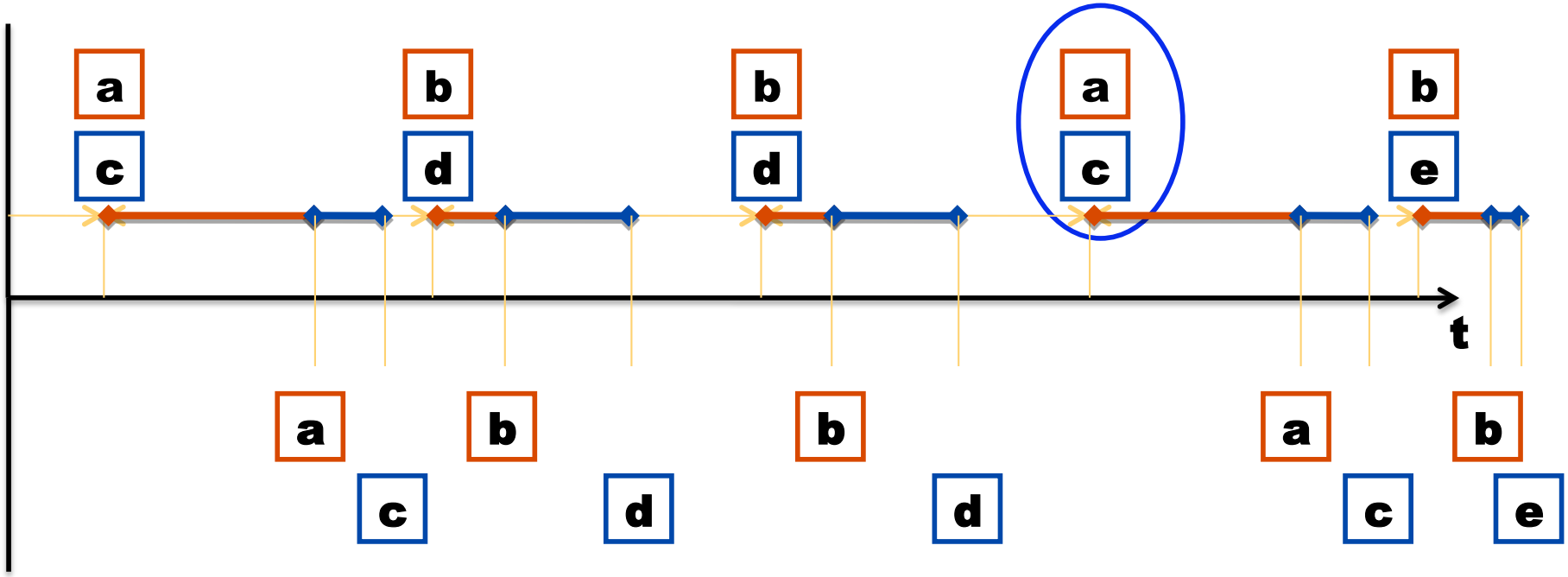



- **scheduling**
  - **periodic activation**
  - **static priority preemptive (SPP)**
  - **T4 higher priority than T1**
- **input dependent execution times**
  - **execution annotated by symbols**

# Scheduling – Example of Symbol Coincidence



# Symbol Coincidence Effect




- response time and jitter of T1 jobs are determined by the content of coinciding T4 symbols!
- composition of load models must provide the *rate of coincidence* of two symbols from different streams
  - must include stochastic task *and job* dependencies

# Symbol Coincidence

- **symbol frequency example**
  - **„a“ 35%, „b“ 65%**
  - **„c“ 25%, „d“ 40%, „e“ 25%, „f“ 10%**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 35% | **a** | = | $P_{1,1}$ | + | $P_{1,2}$ | + | $P_{1,3}$ | + | $P_{1,4}$ |
| | | | + | | + | | + | | + |
| 65% | **b** | = | $P_{2,1}$ | + | $P_{2,2}$ | + | $P_{2,3}$ | + | $P_{2,4}$ |
| | | | = | | = | | = | | = |
| | | | **f** | | **c** | | **d** | | **e** |
| | | | 10% | | 25% | | 40% | | 25% |

# Independence Assumption

| | | | | | |
|---|---|---|---|---|---|
| 35% | **a** | 3.5% | 8.75% | 14% | 8.75% |
| 65% | **b** | 6.5% | 16.25% | 26% | 16.25% |
| | | **f** | **c** | **d** | **e** |
| | | 10% | 25% | 40% | 25% |

# Independence vs. No Assumption

| | | | | | |
|---|---|---|---|---|---|
| 35% | a | 3.5% | 8.75% | 14% | 8.75% |
| 65% | b | 6.5% | 16.25% | 26% | 16.25% |
| | | f | c | d | e |
| **independence** | | 10% | 25% | 40% | 25% |

| | | | | | |
|---|---|---|---|---|---|
| 35% | a | 0-10% | 0-25% | 0-35% | 0-25% |
| 65% | b | 0-10% | 0-25% | 0-40% | 0-25% |
| | | f | c | d | e |
| **no assumption** | | 10% | 25% | 40% | 25% |

# Consequences for Composition

- **response time distribution dependencies do not only come from functional dependencies but also from *non functional platform effects***
- **so, according to de Alfaro et al. resource scheduling must be restricted to isolated platform usage (e.g. in TDMA) – unless all effects explicitely modeled**
  - **counter the required effect of higher efficiency**
- **however, even if unknown, platform dependency effects can be captured by probability bounds!**

⇒ ***can allow composition using scheduling with unknown side effects***

⇒ ***BUT: need composition operators for scheduling algorithms on bounded probabilities***

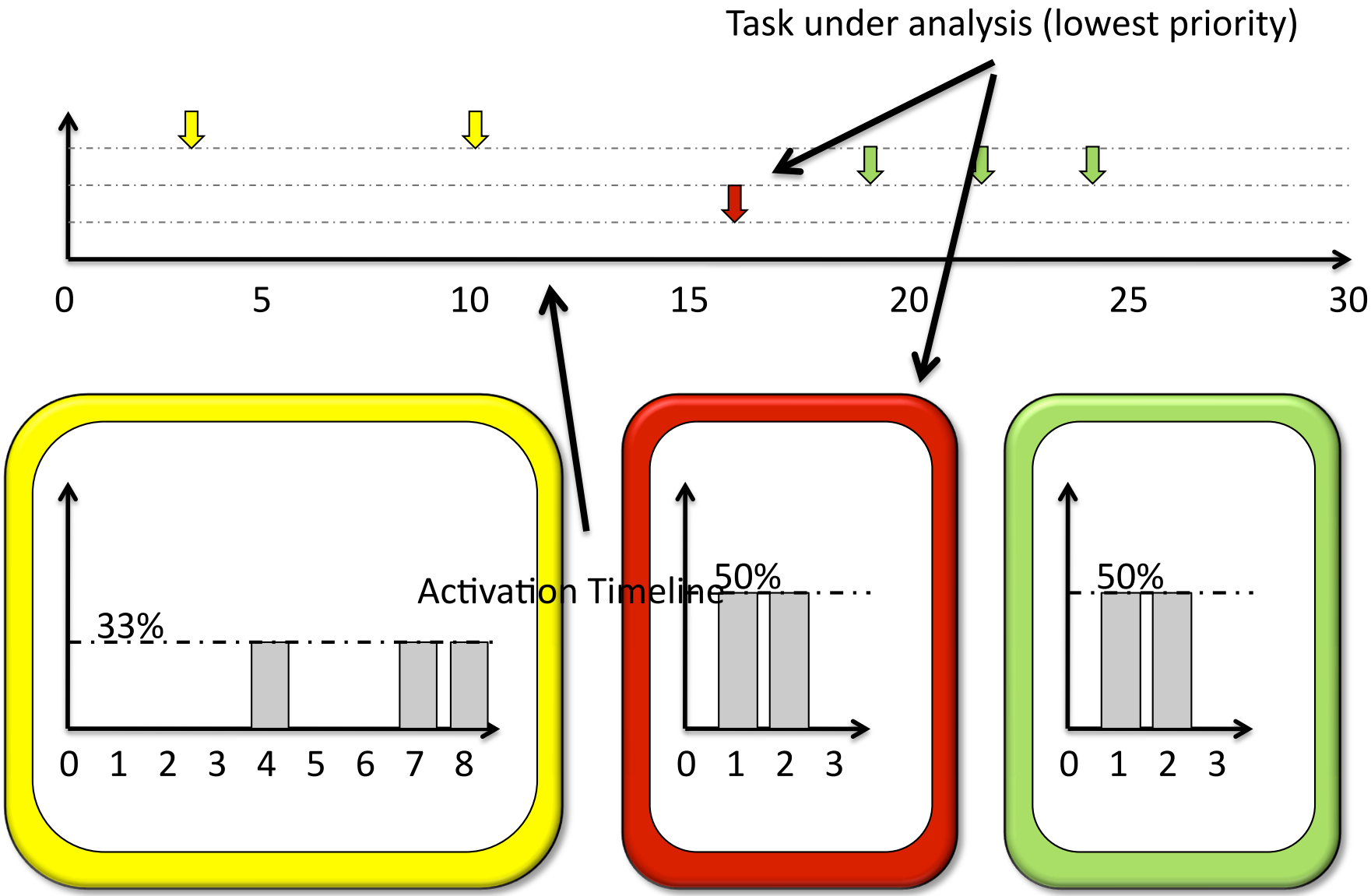
# Overview

- motivation
- probabilistic models in systems integration
- scheduling analysis using probabilistic bounds
- combining WC intervals and probabilistic bounds
- conclusion

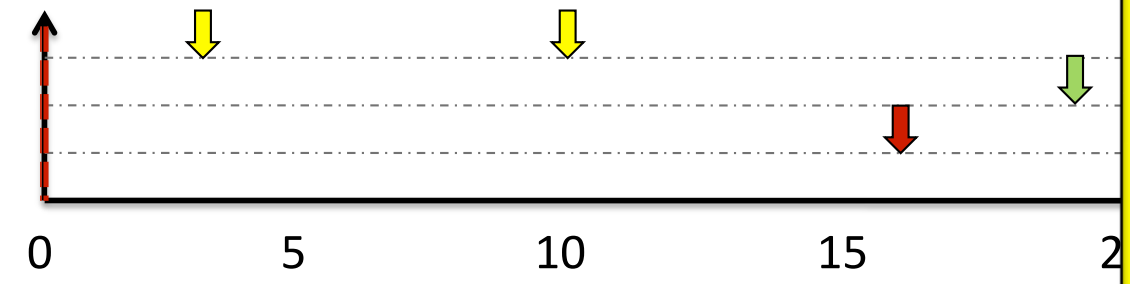
# Application to Algorithm of Diaz (2005 et al.)

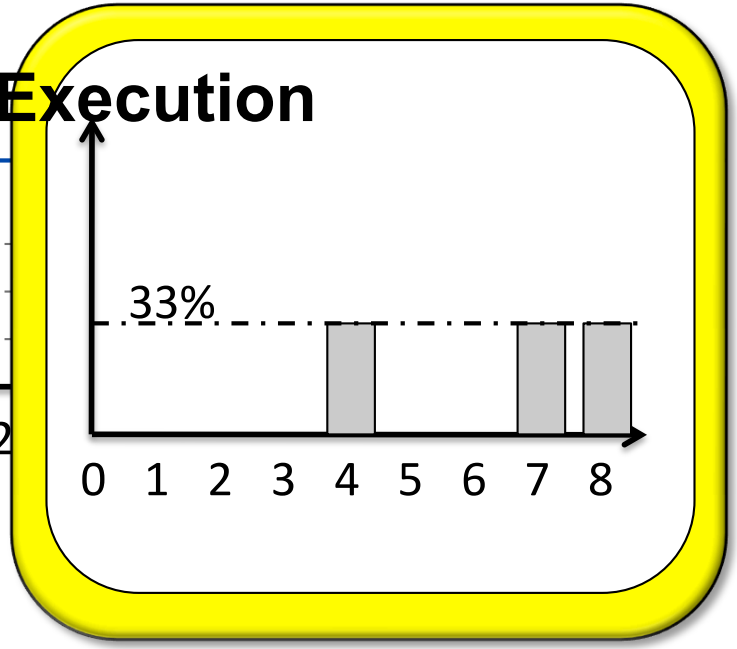
- **Diaz scheduling algorithm and task model**
  - **static priority preemptive scheduling**
  - **periodic task activation patterns (hyperperiod exists)**
  - **stochastic job execution times**
  - **load backlog beyond hyperperiod permitted (U>1, uses Markov analysis)**
  - ***independent tasks, independent jobs***
- **modified task model (this talk)**
  - **dependent tasks and jobs**
  - **(currently) no load backlog beyond hyperperiods (U<=1)**
  - **stochastic task activation times (omitted in the following slides)**

# Example System

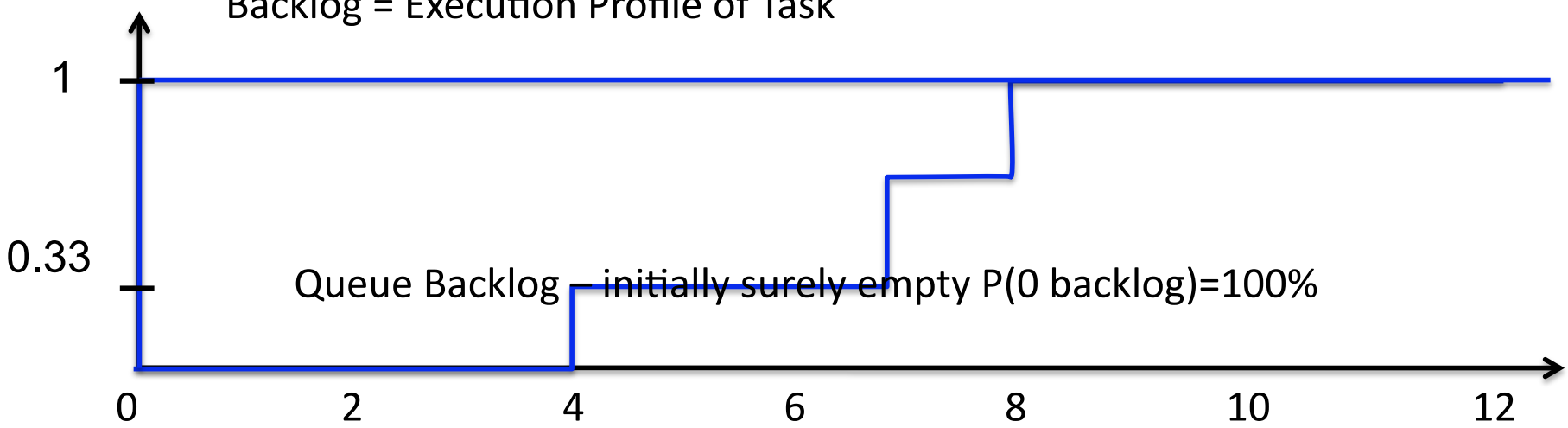


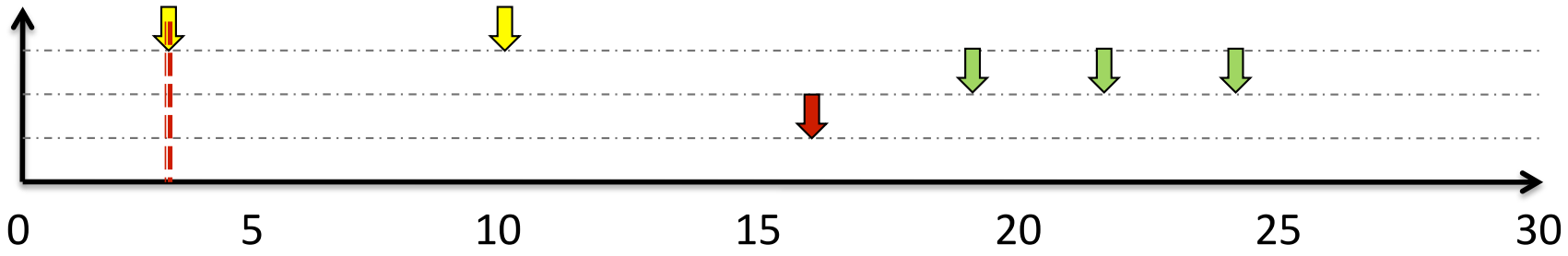
# Example – First Task Execution



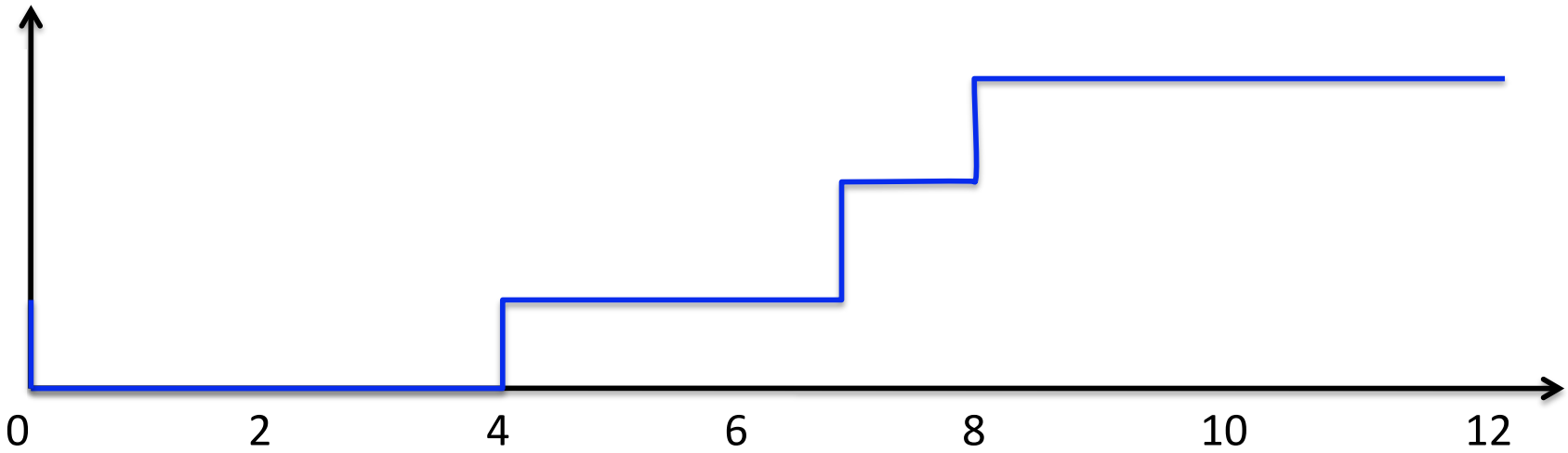



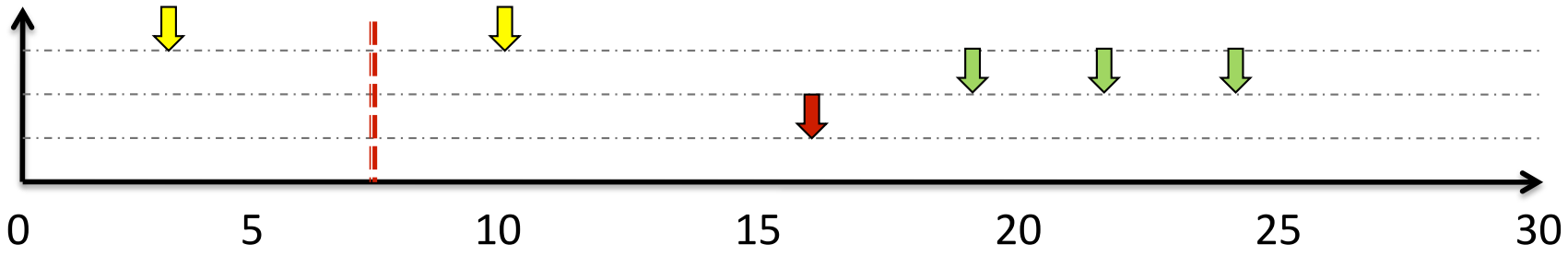

First Activation
– empty Queue
Backlog = Execution Profile of Task



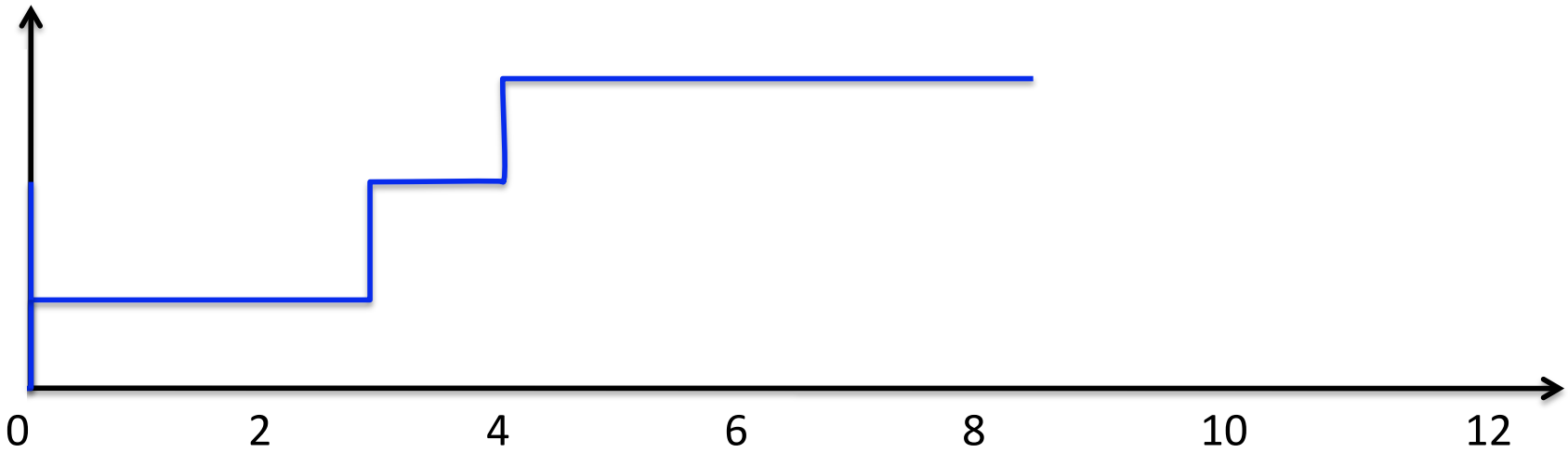


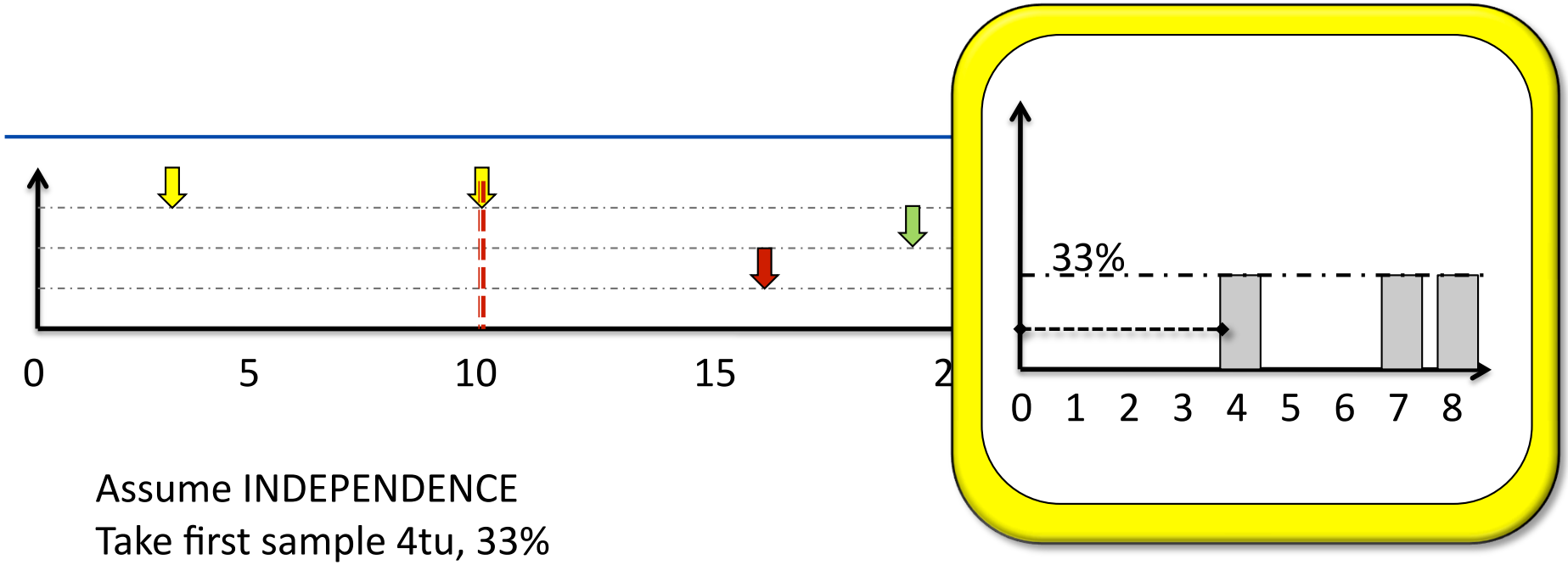

Time passes – Queue shrinks



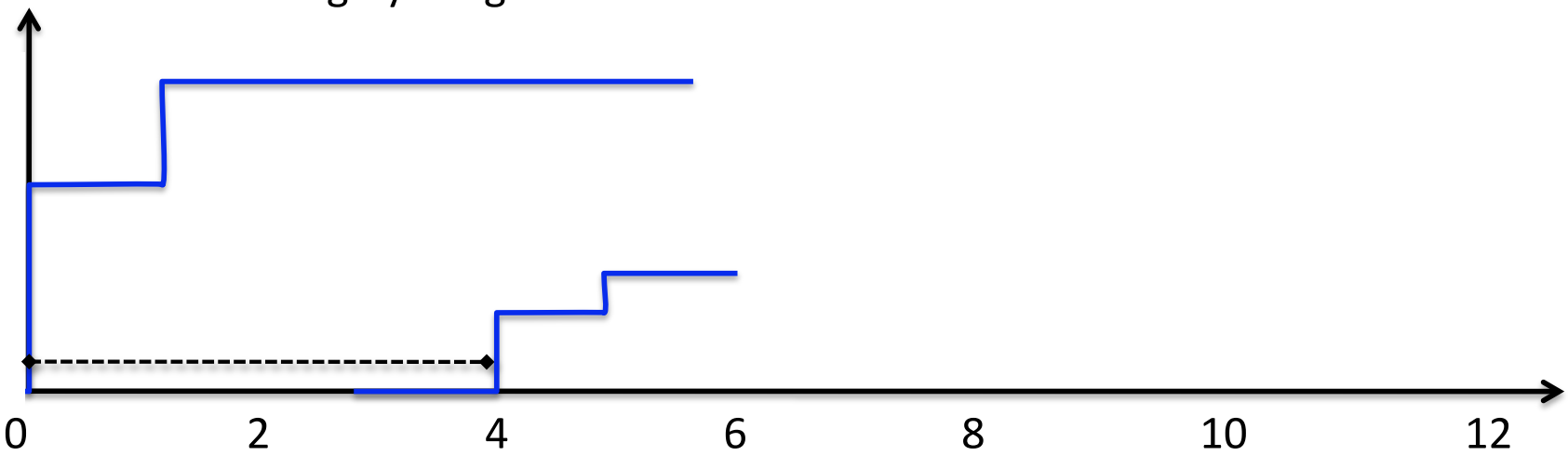


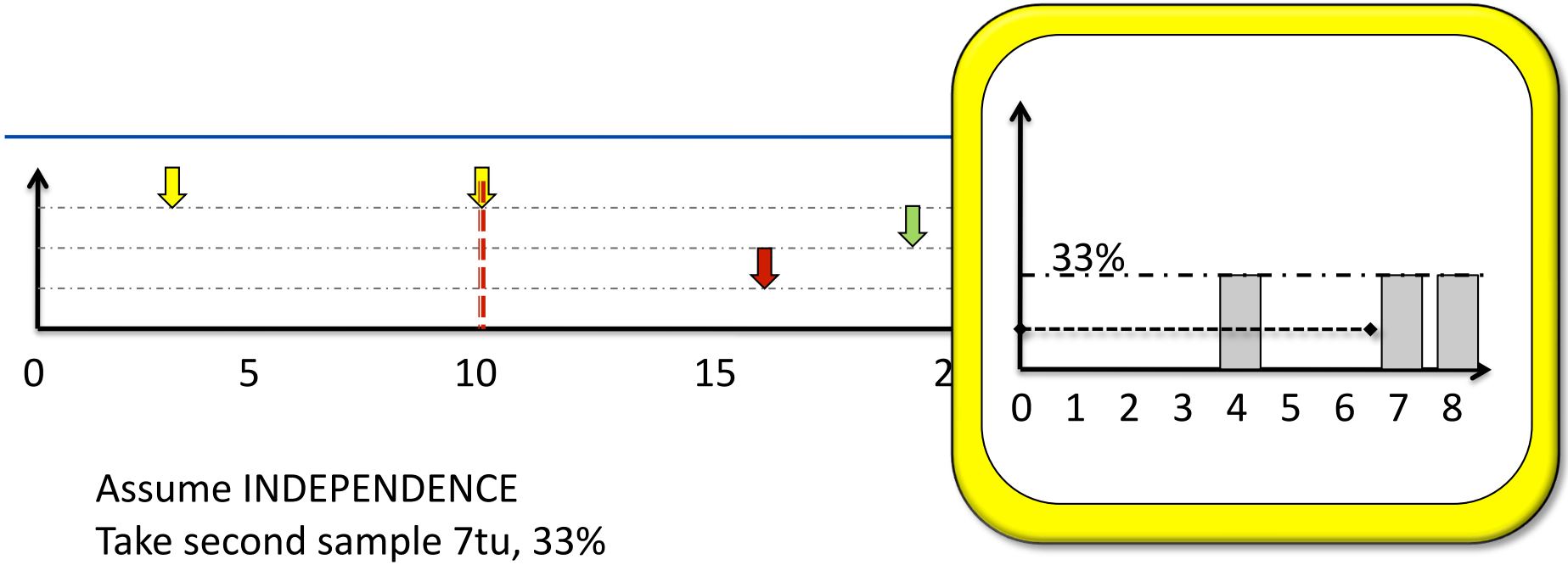

„Negative“ backlog accumulates at 0



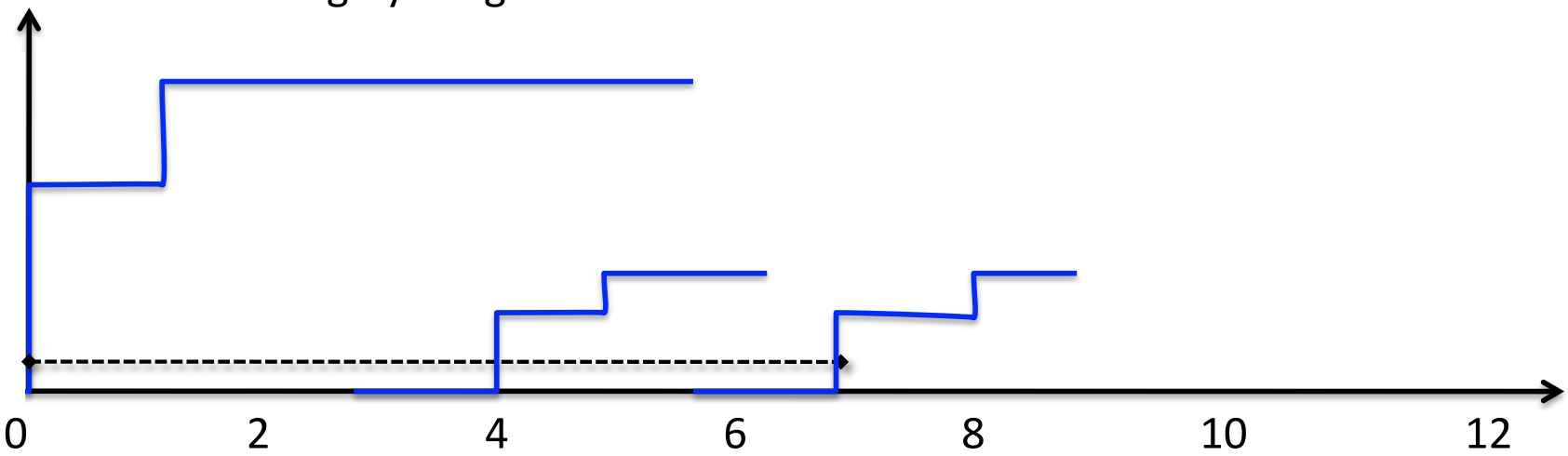


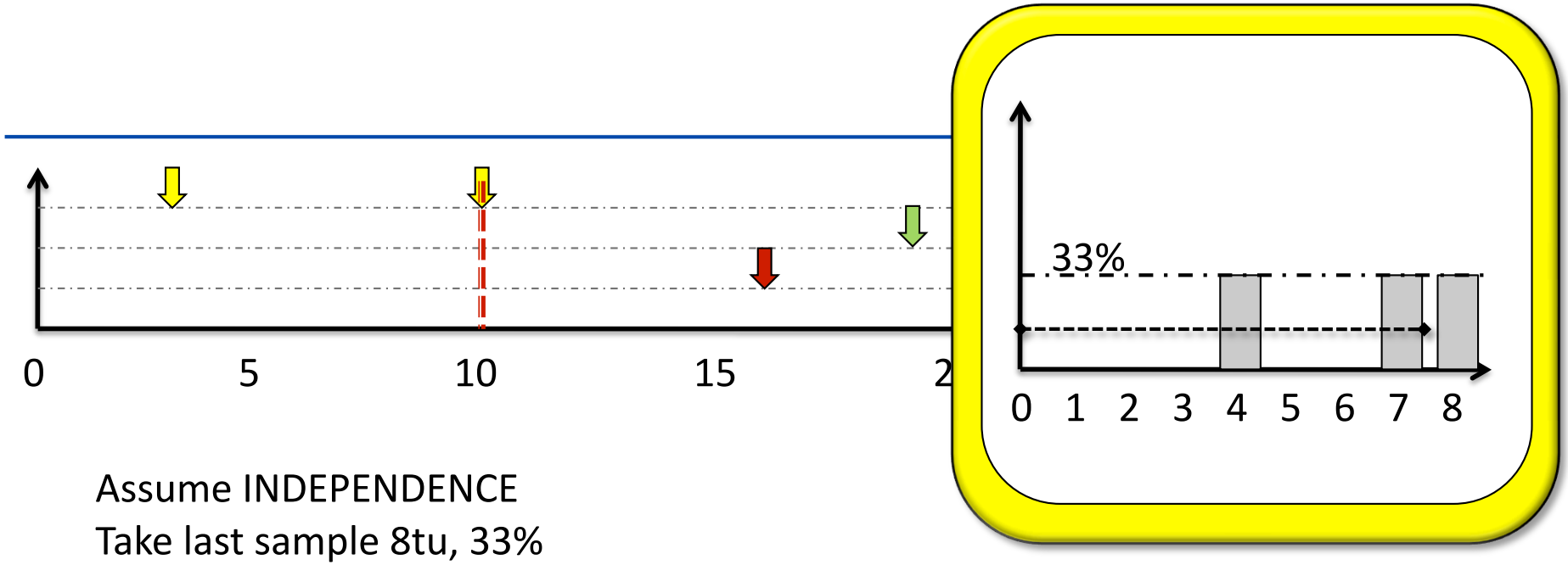

Assume INDEPENDENCE
Take first sample 4tu, 33%
Shift backlog by the new value: 4tu
Scale Backlog by weight: 33%



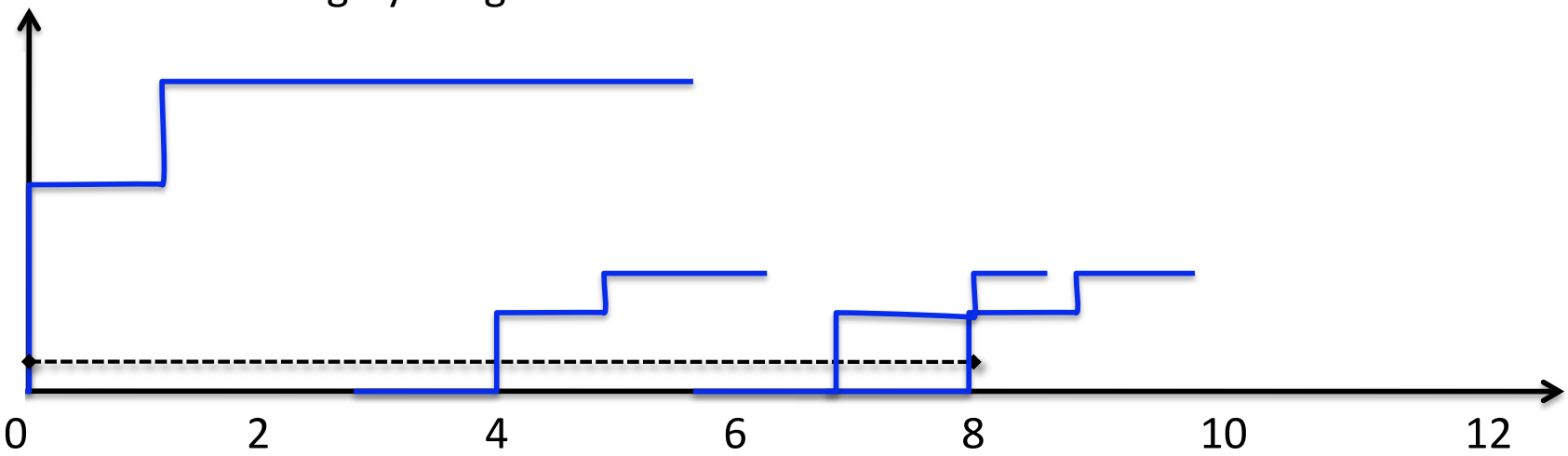


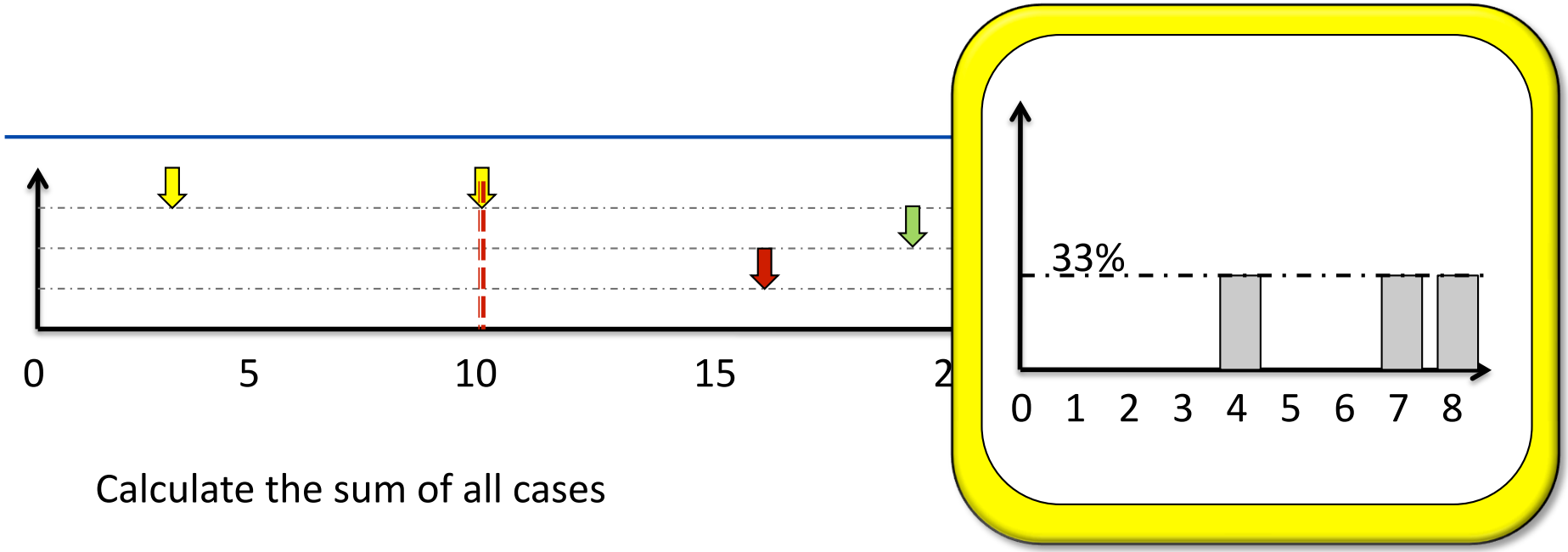

Assume INDEPENDENCE
Take second sample 7tu, 33%
Shift backlog by the new value: 7tu
Scale Backlog by weight: 33%





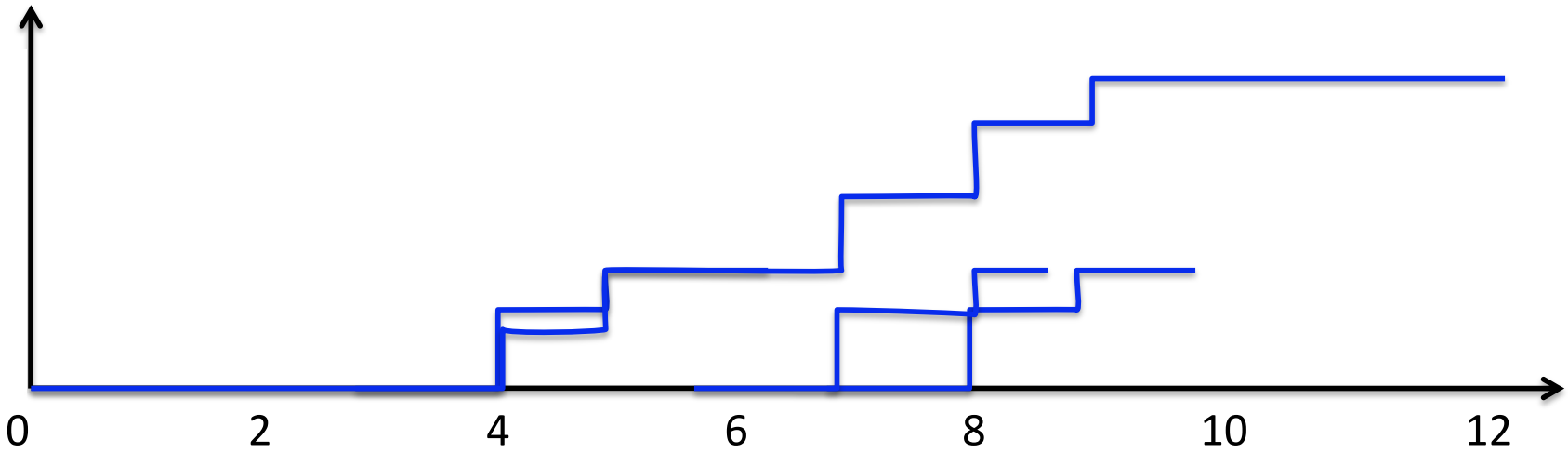

Assume INDEPENDENCE
Take last sample 8tu, 33%
Shift backlog by the new value: 8tu
Scale Backlog by weight: 33%

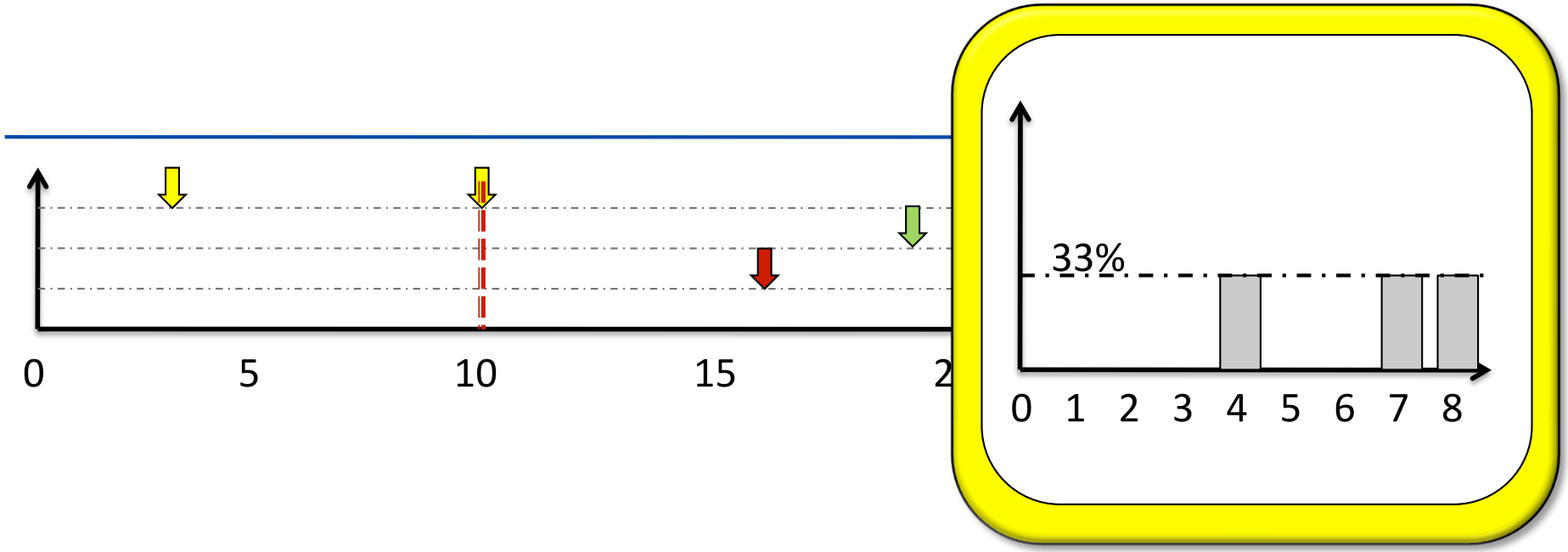


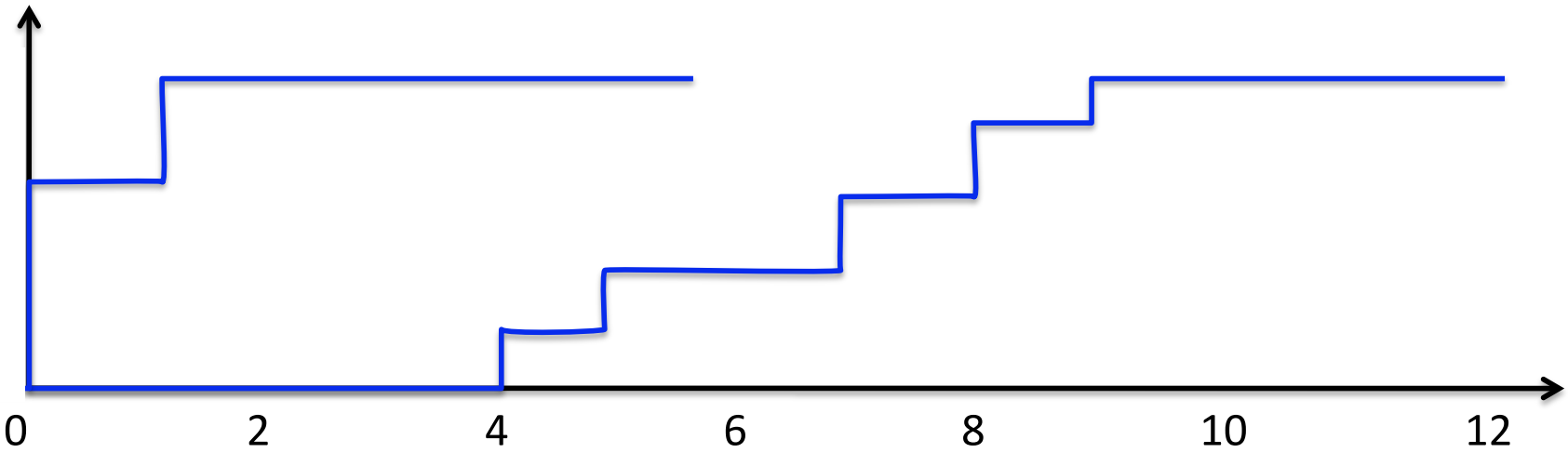


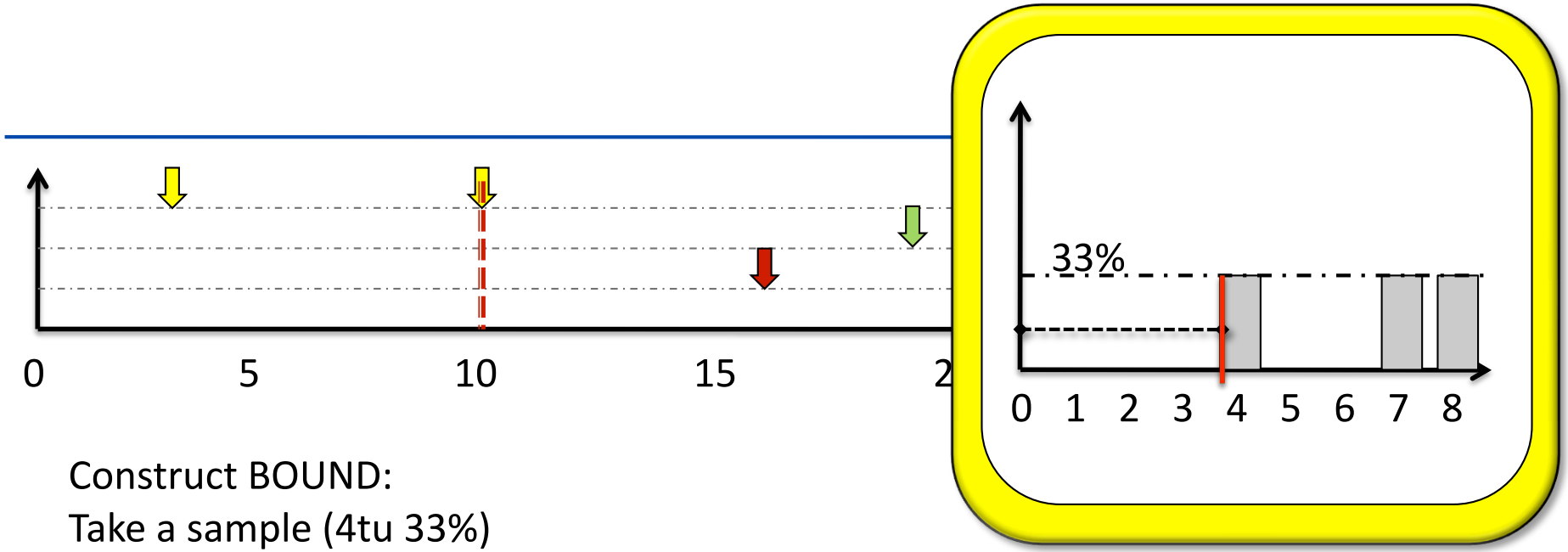

Calculate the sum of all cases

Done for INDEPENDENCE

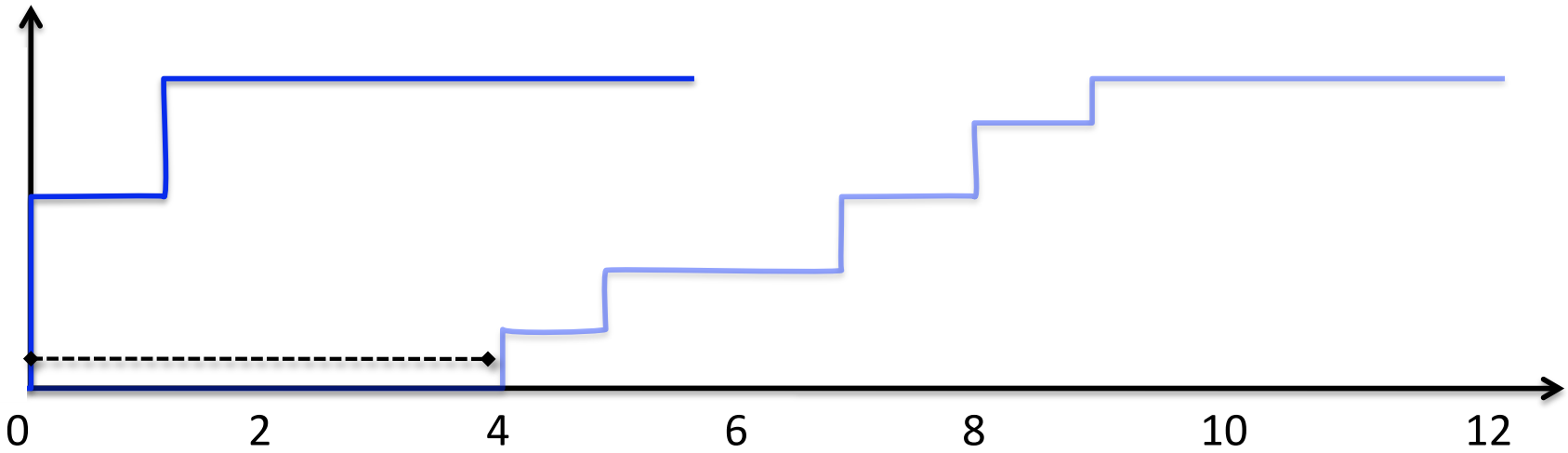


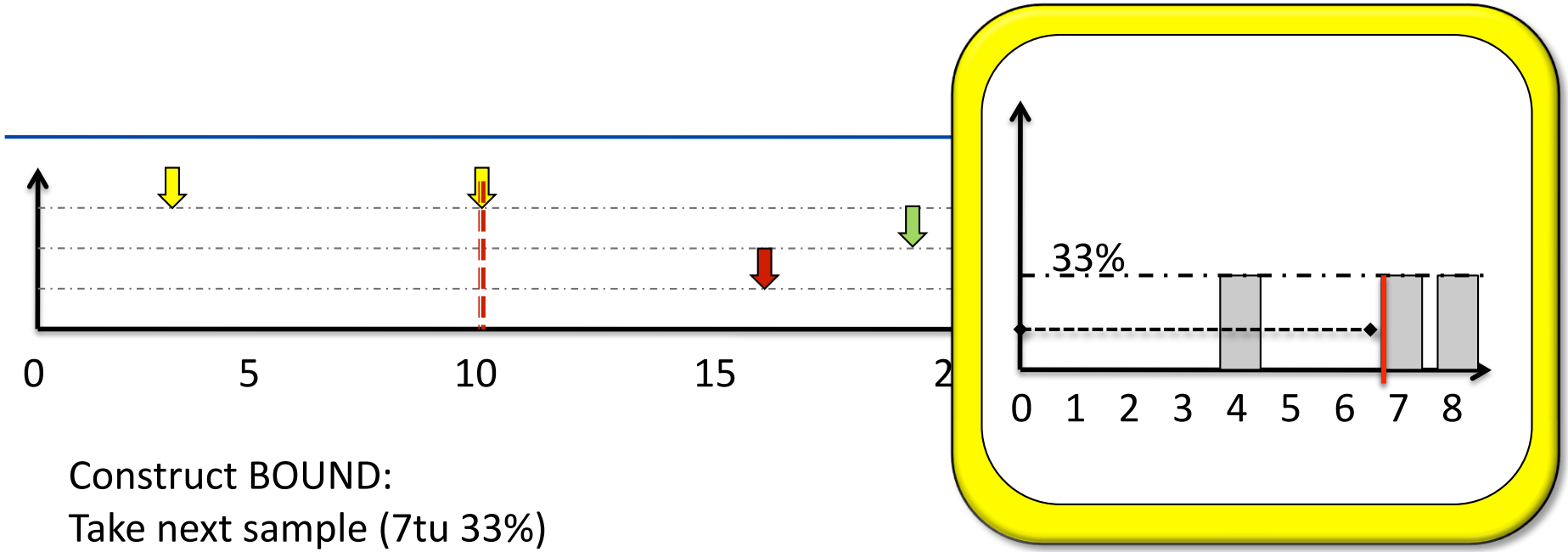

0
5
10
15
2
33%
0 1 2 3 4 5 6 7 8


0
2
4
6
8
10
12

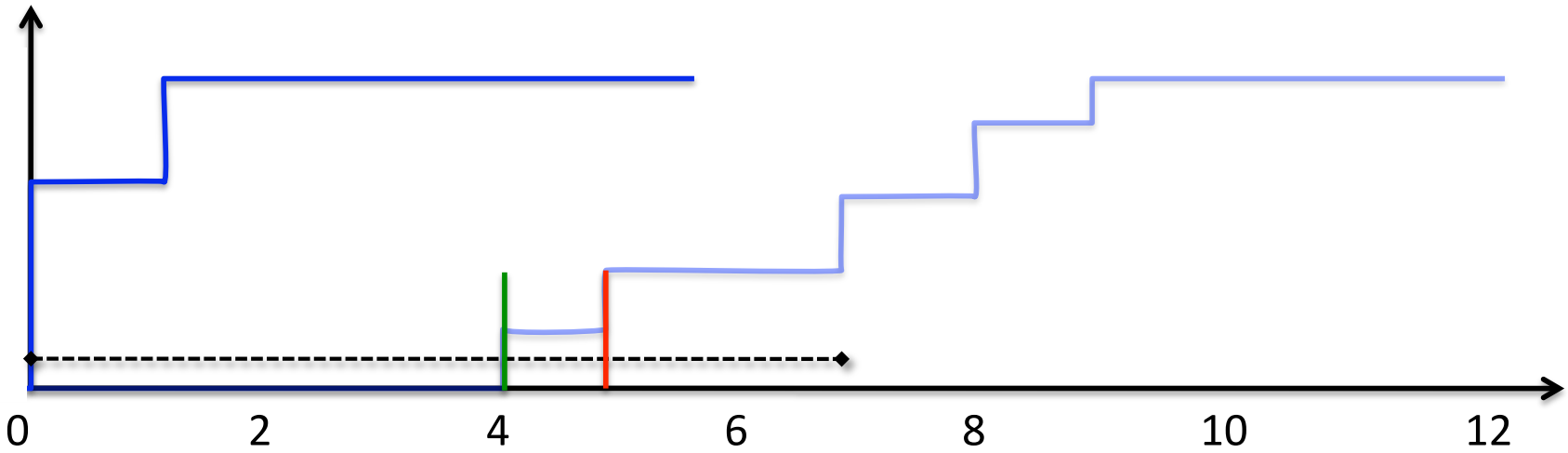


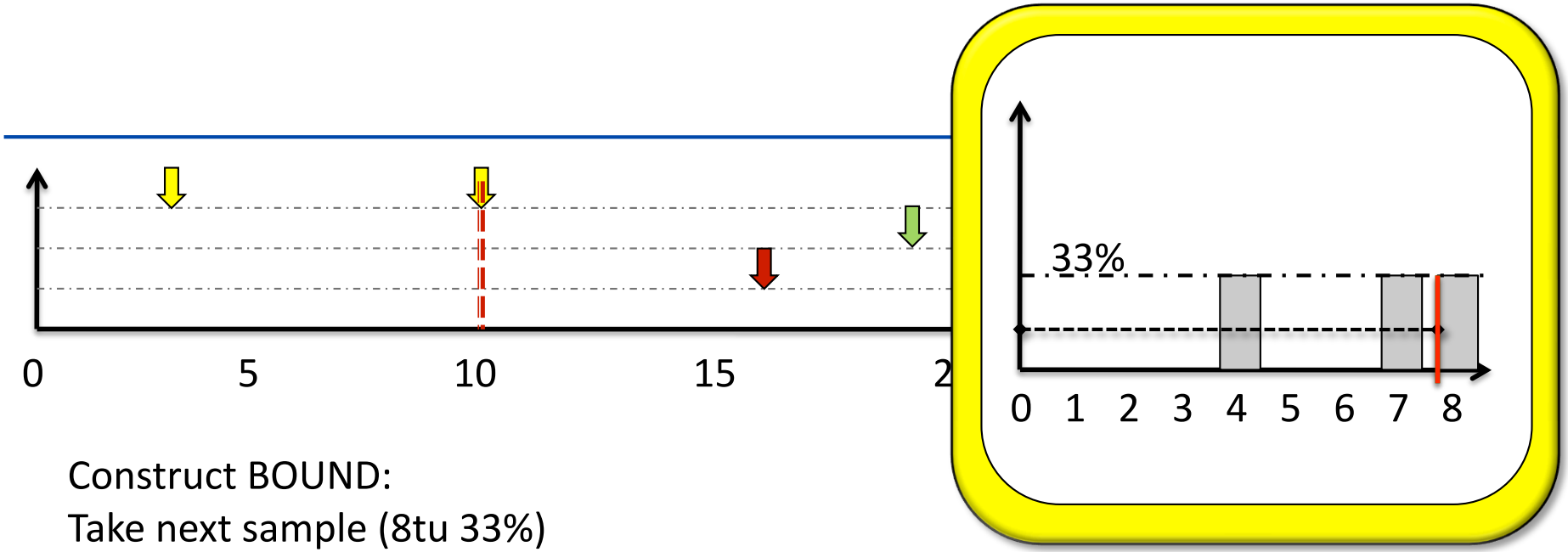

Construct BOUND:
Take a sample (4tu 33%)
Align it with best/worst case of the backlog
Shift it by the delay: 4tu



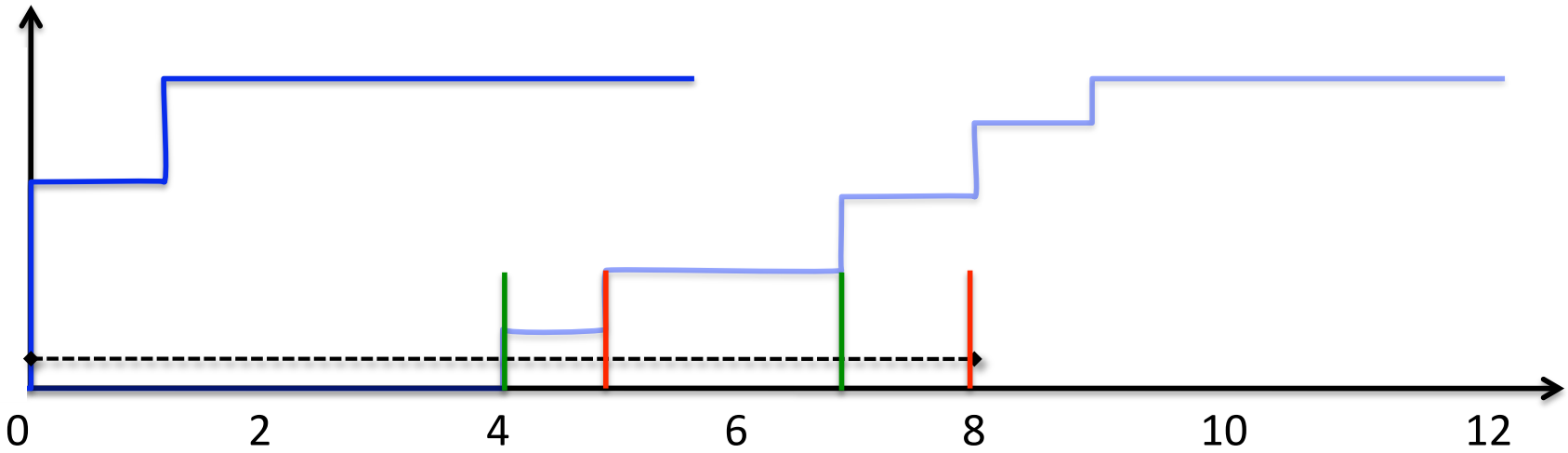


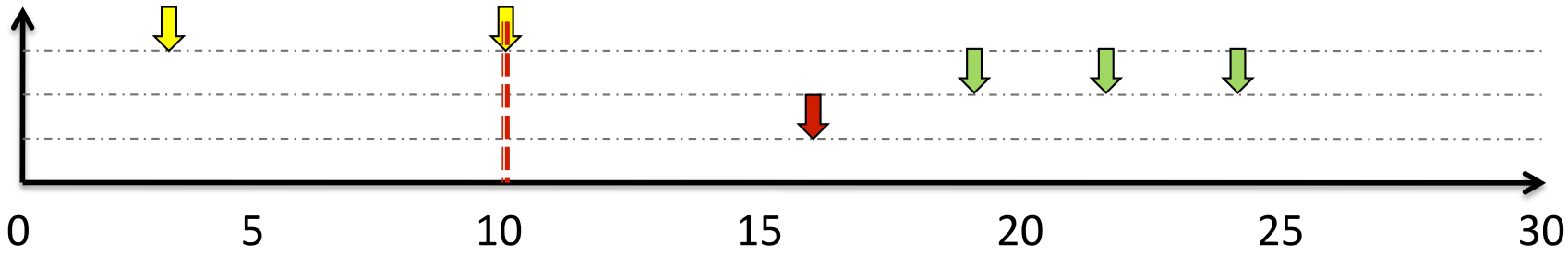

Construct BOUND:
Take next sample (7tu 33%)
Align it with best/worst case of the backlog
Shift it by the delay: 7tu





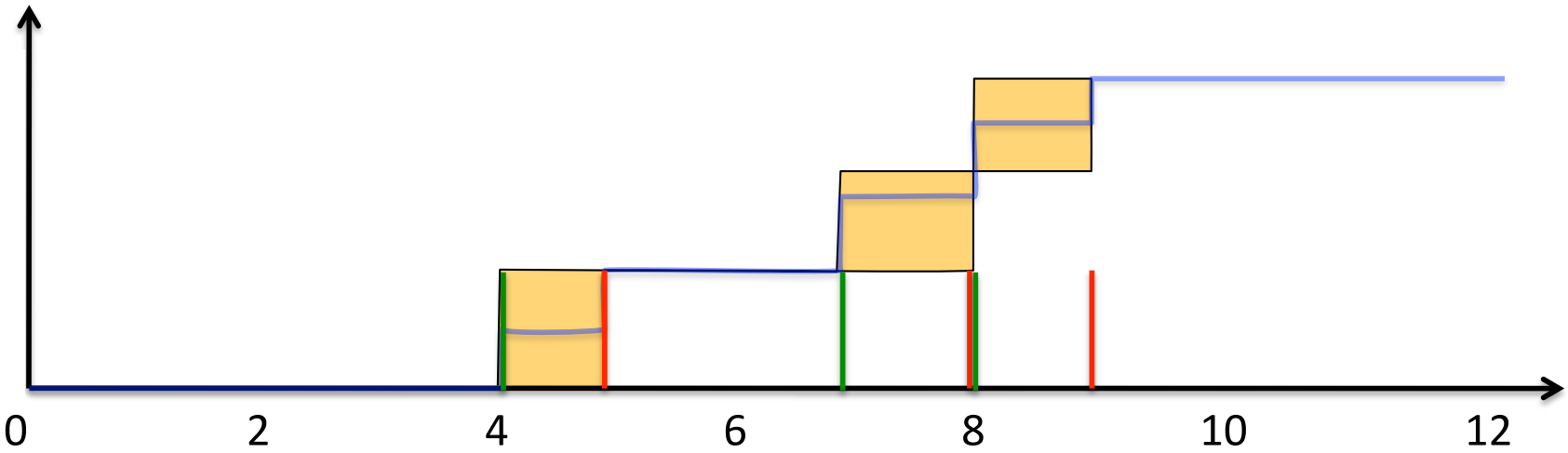

Construct BOUND:
Take next sample (8tu 33%)
Align it with best/worst case of the backlog
Shift it by the delay: 8tu

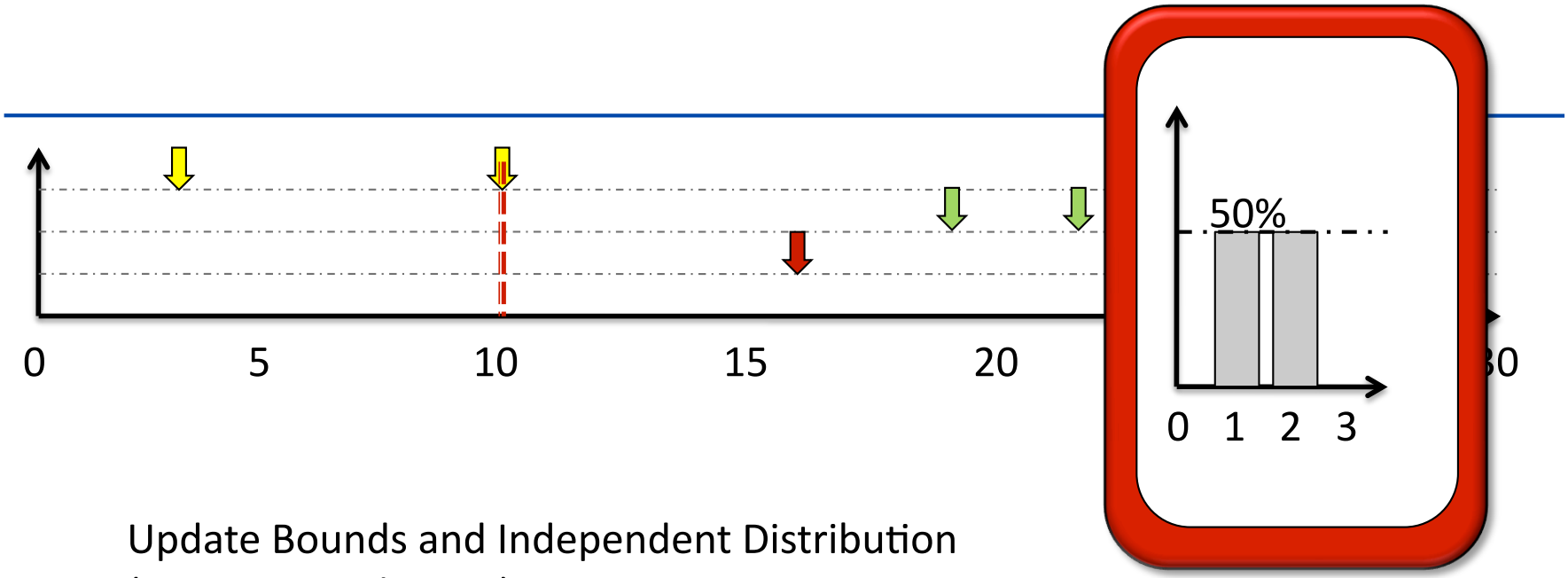




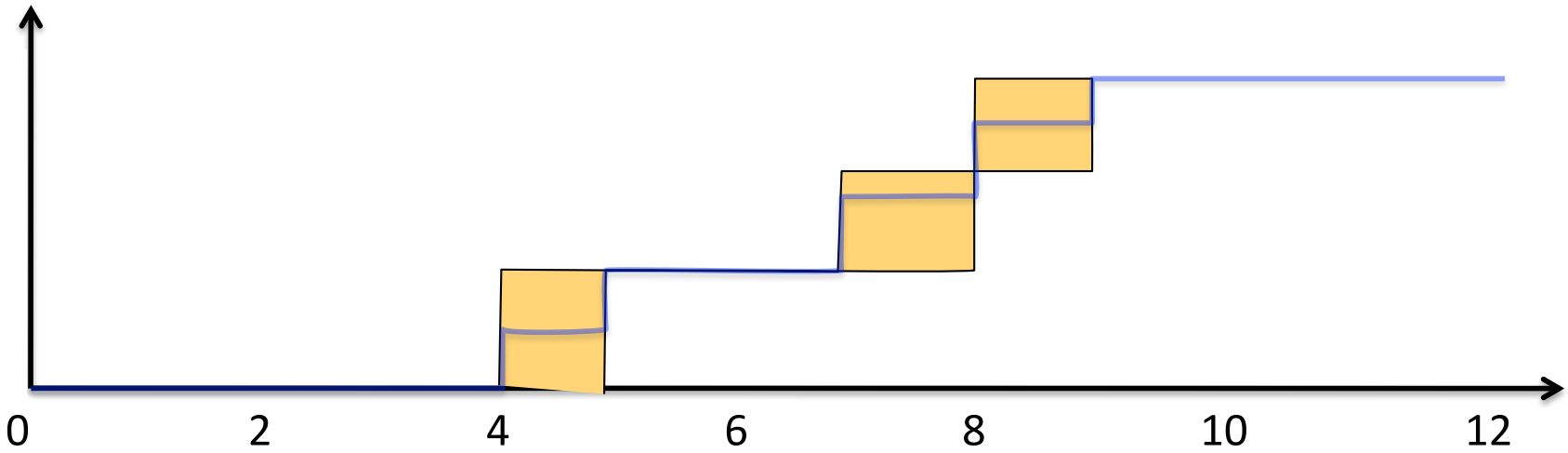

Calculate the sum of all cases

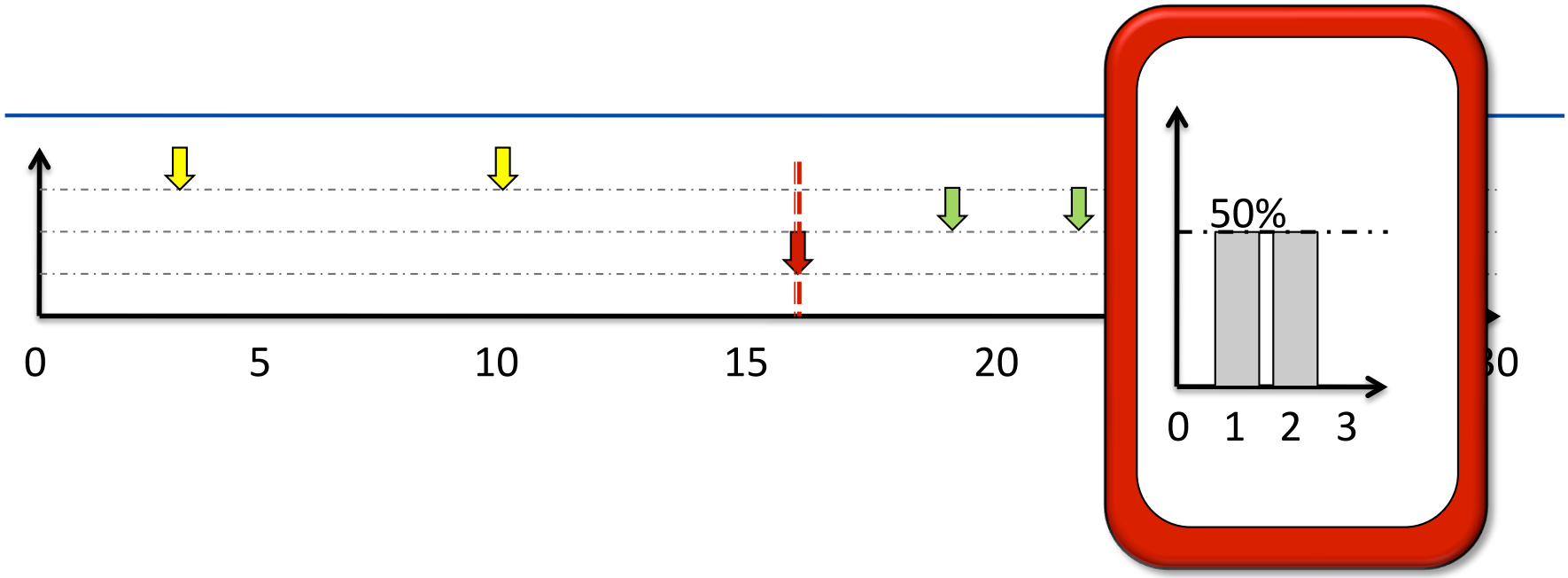
Done for BOUNDS

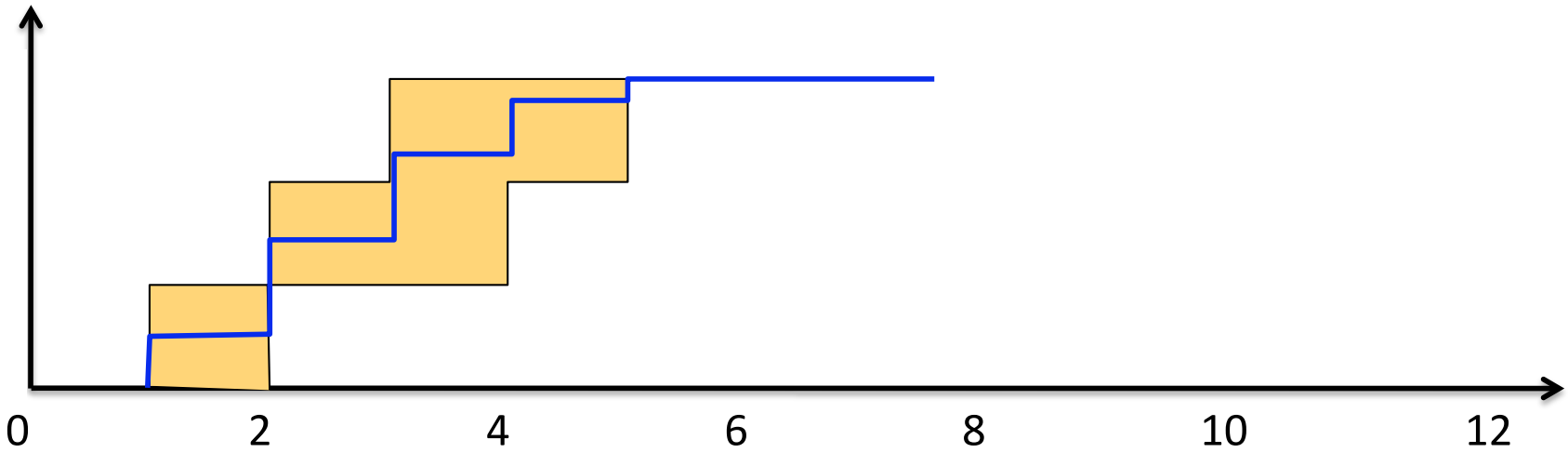


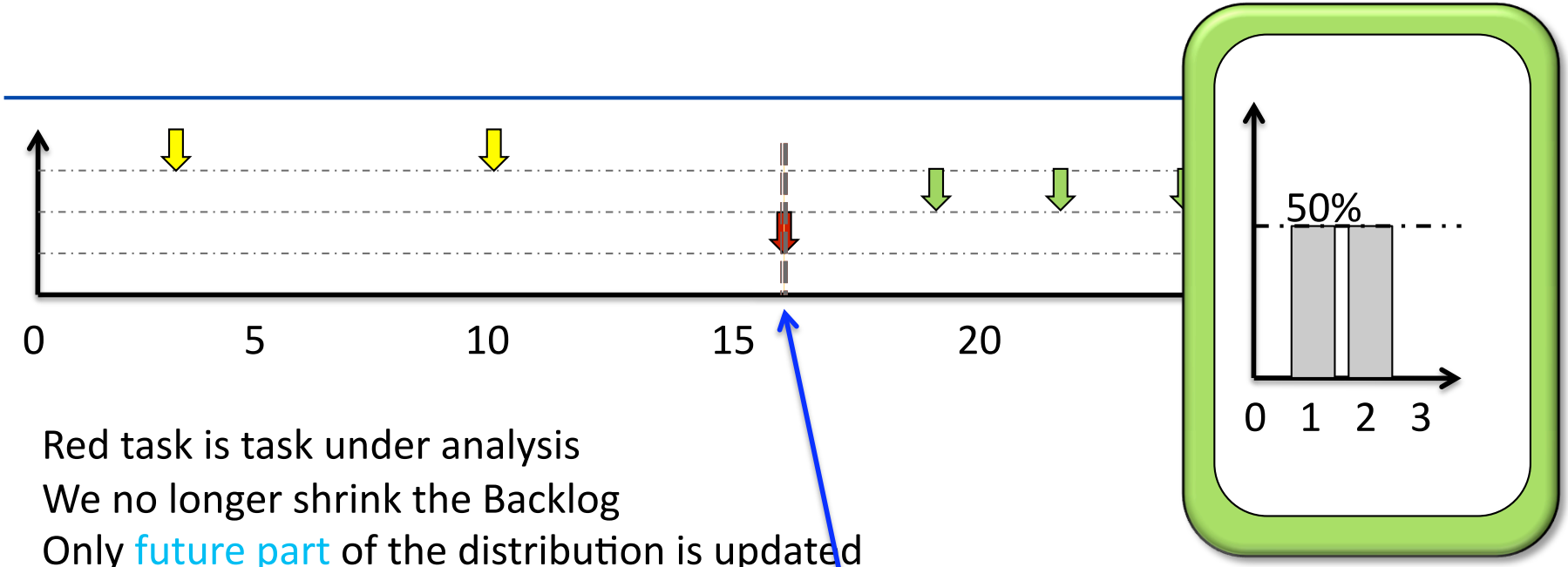



Update Bounds and Independent Distribution
(now in a single step)

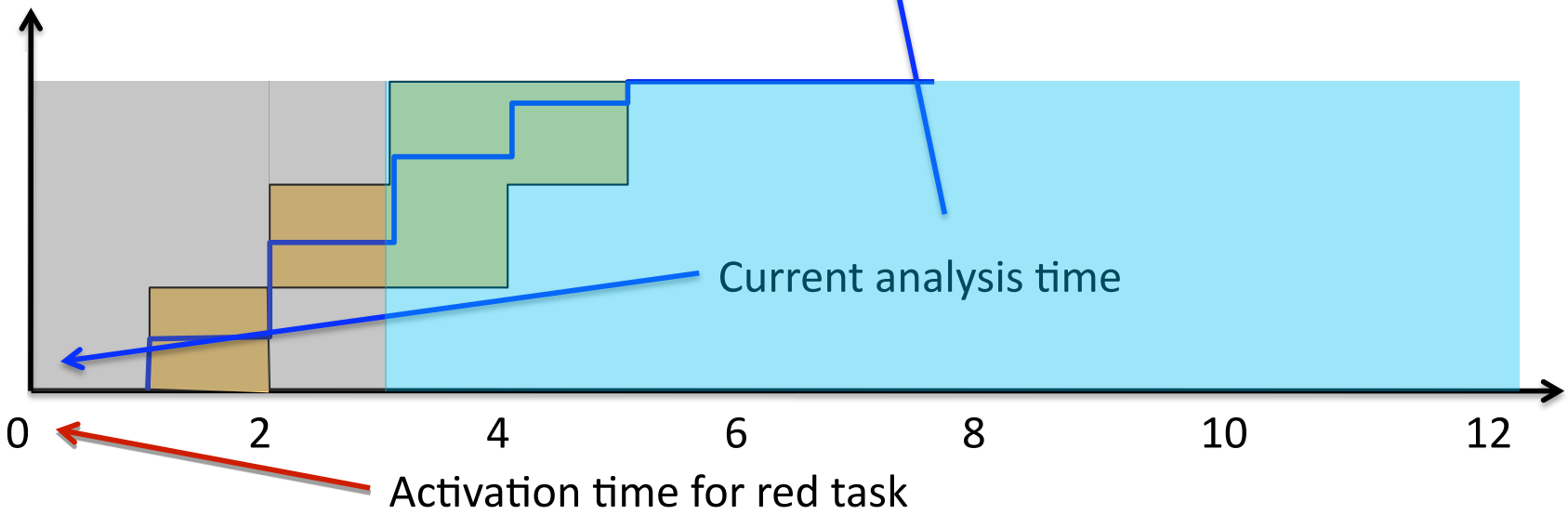


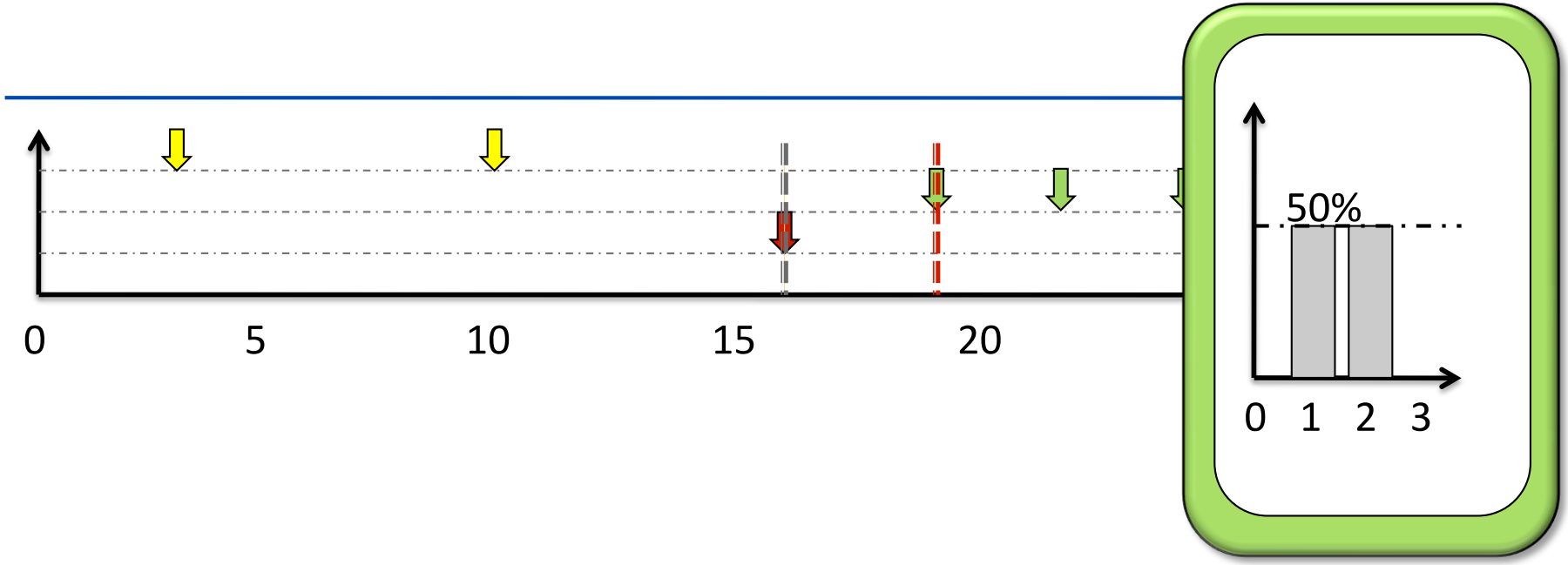

0
5
10
15
20
30
50%
0 1 2 3

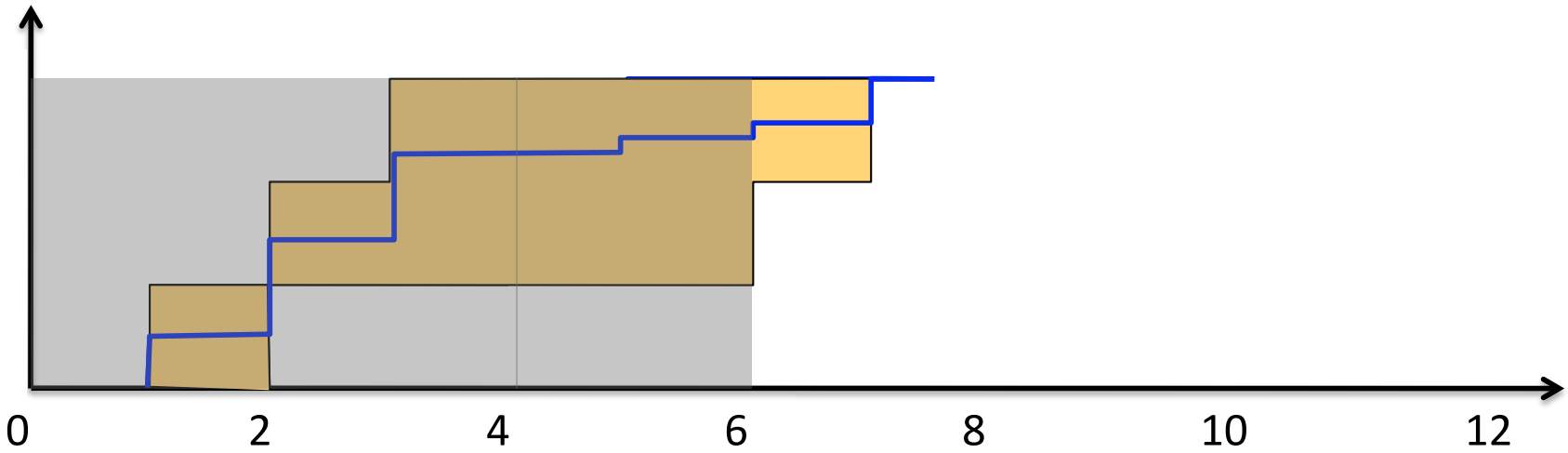

0
2
4
6
8
10
12

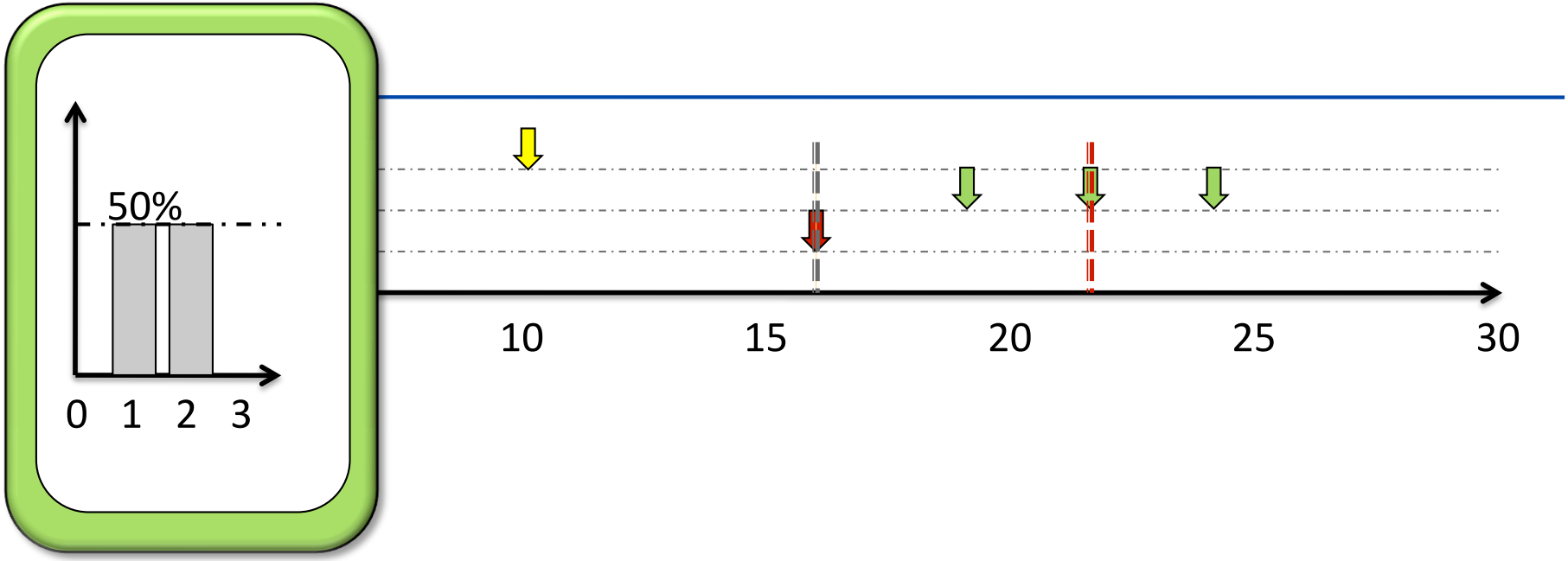


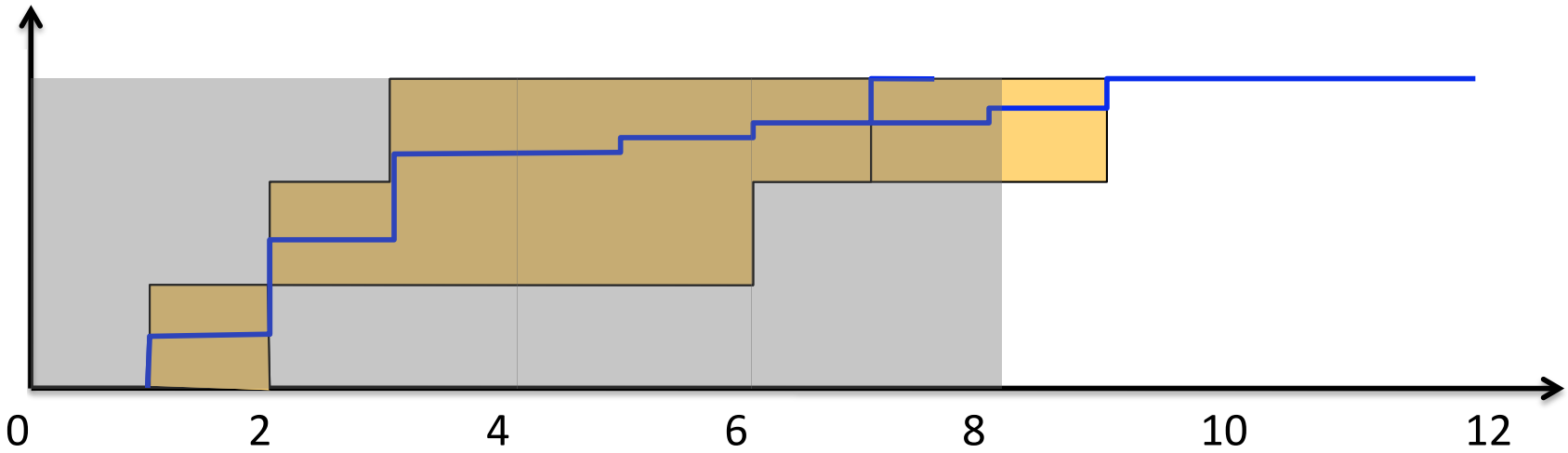

Red task is task under analysis
We no longer shrink the Backlog
Only future part of the distribution is updated

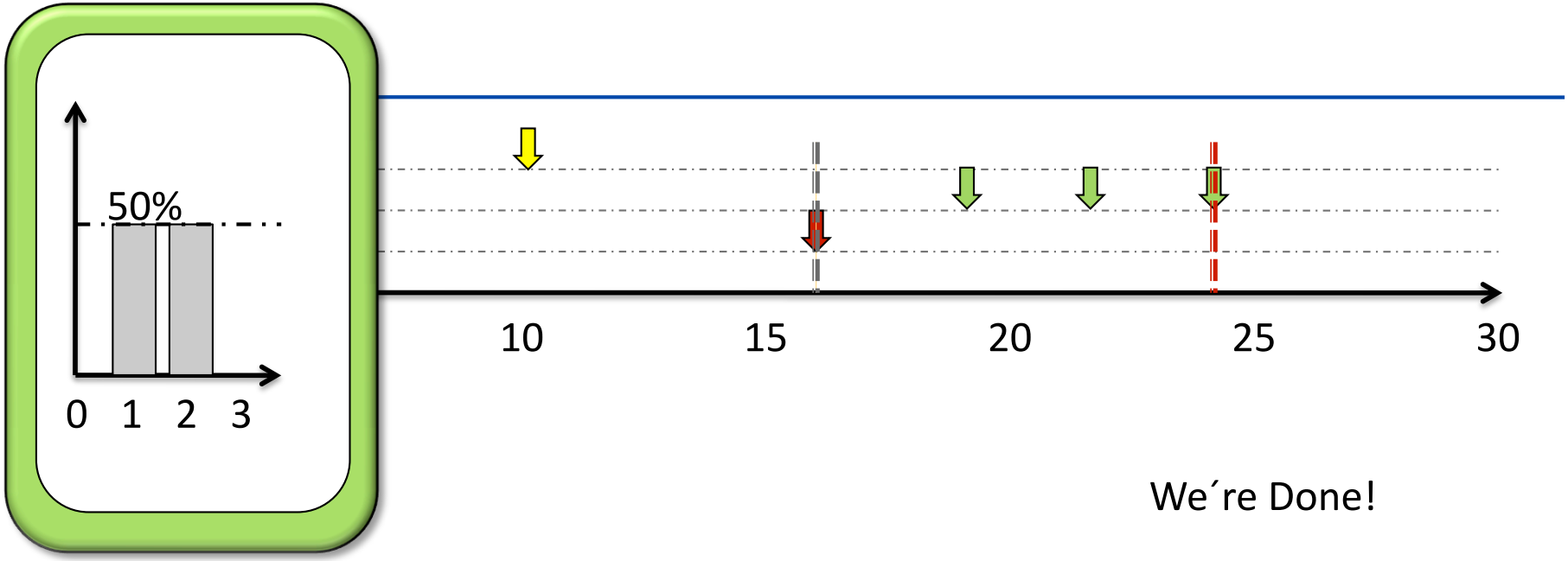



0
5
10
15
20
50%
0
1
2
3

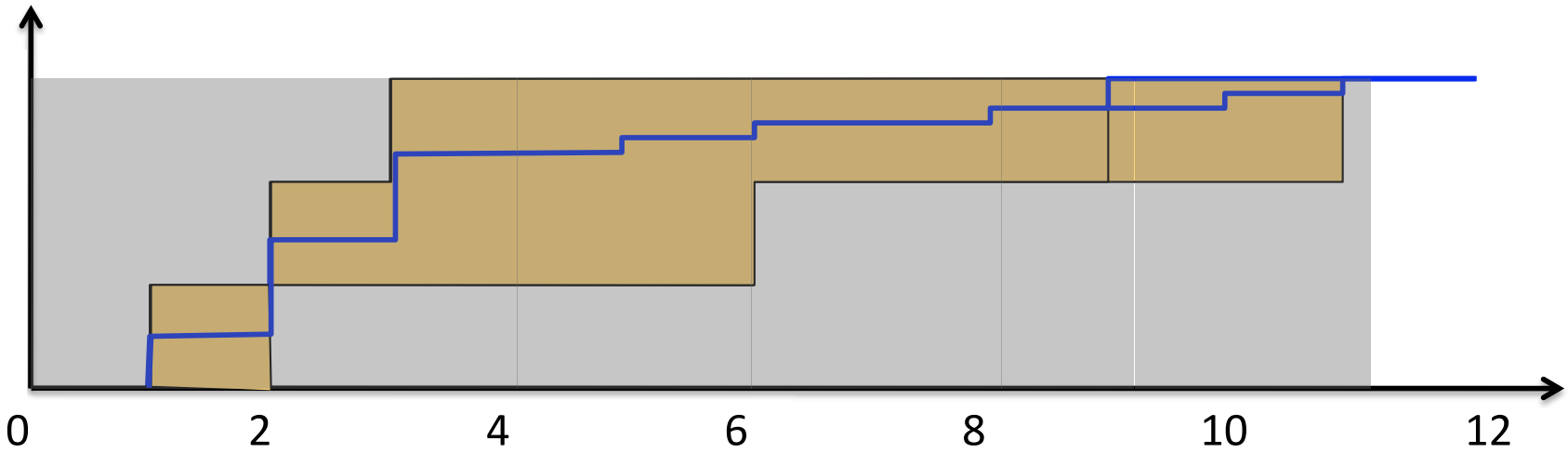

0
2
4
6
8
10
12


50%
0 1 2 3
10
15
20
25
30

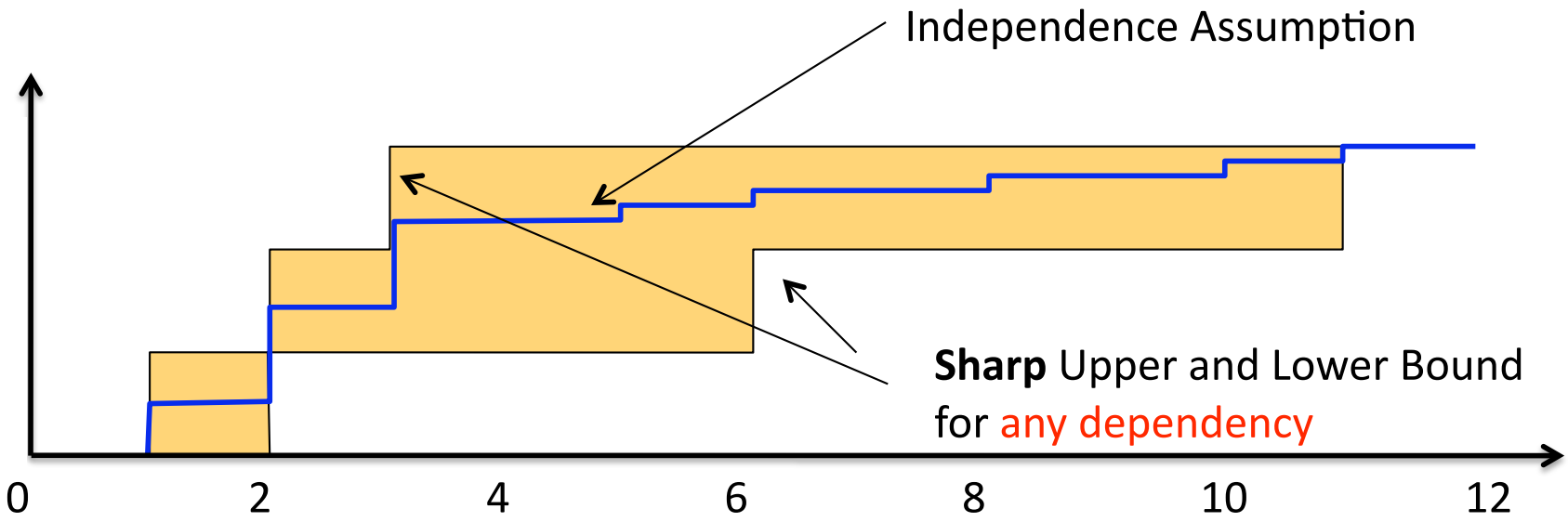

0
2
4
6
8
10
12



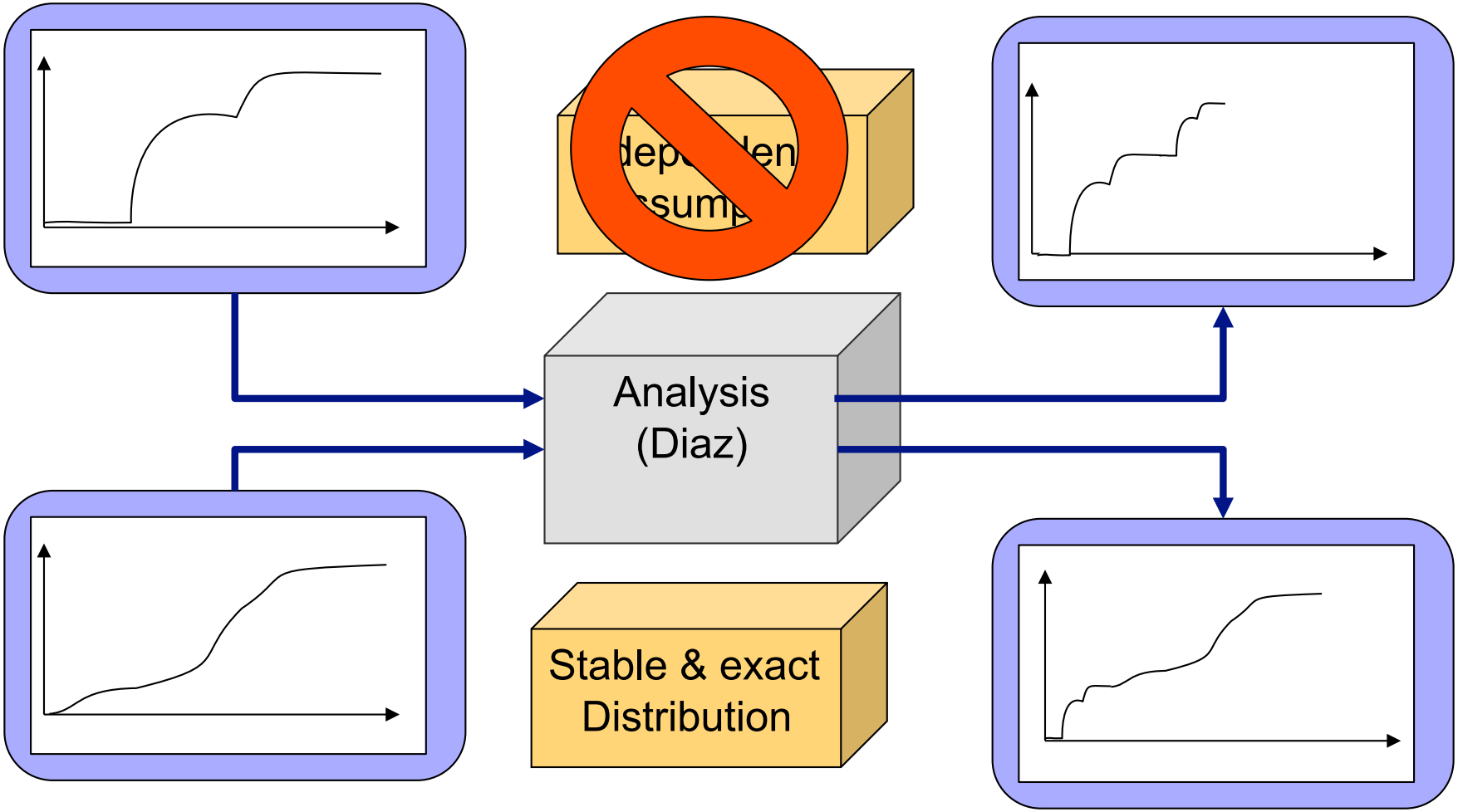

We´re Done!



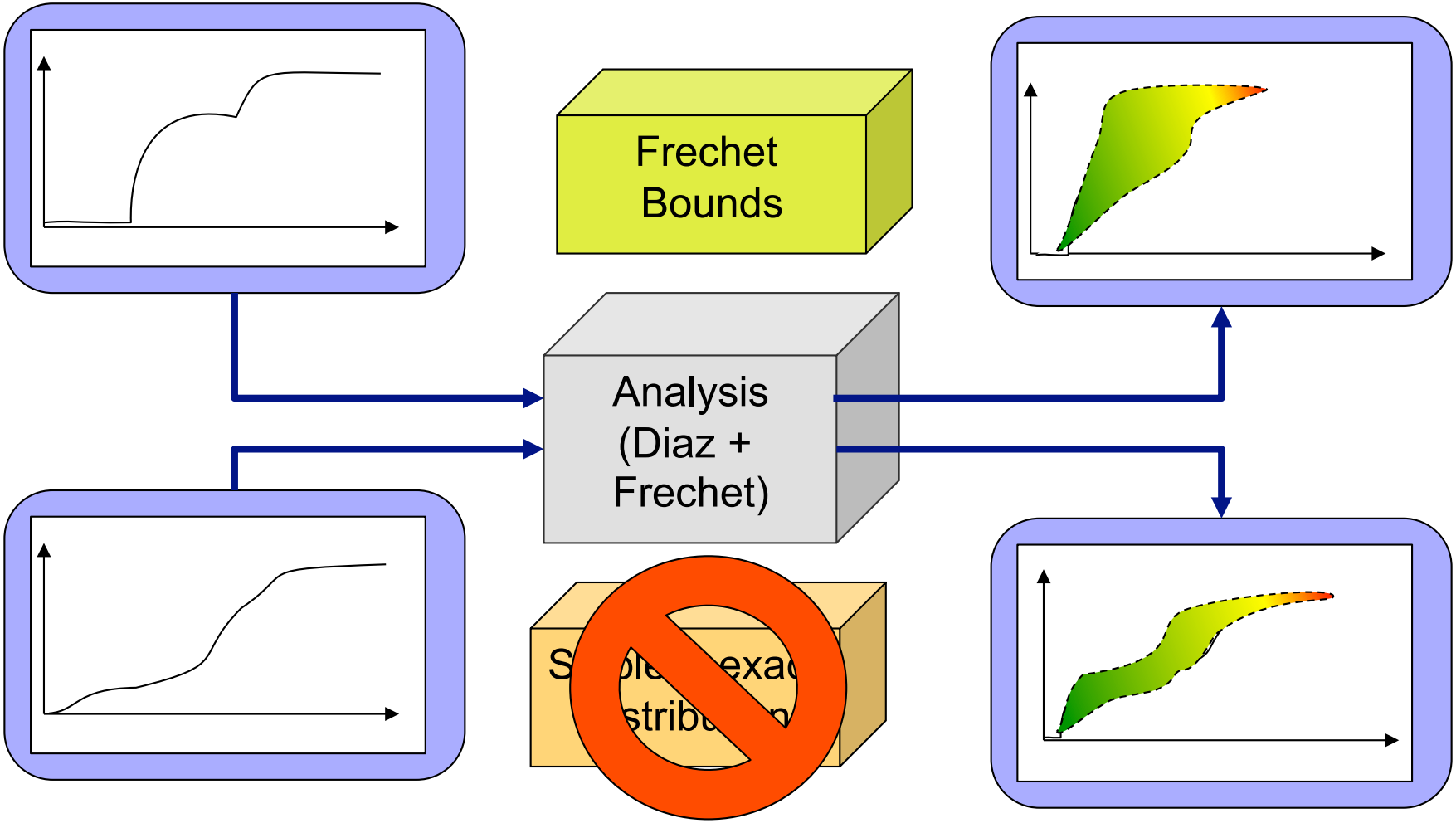
# Final Result



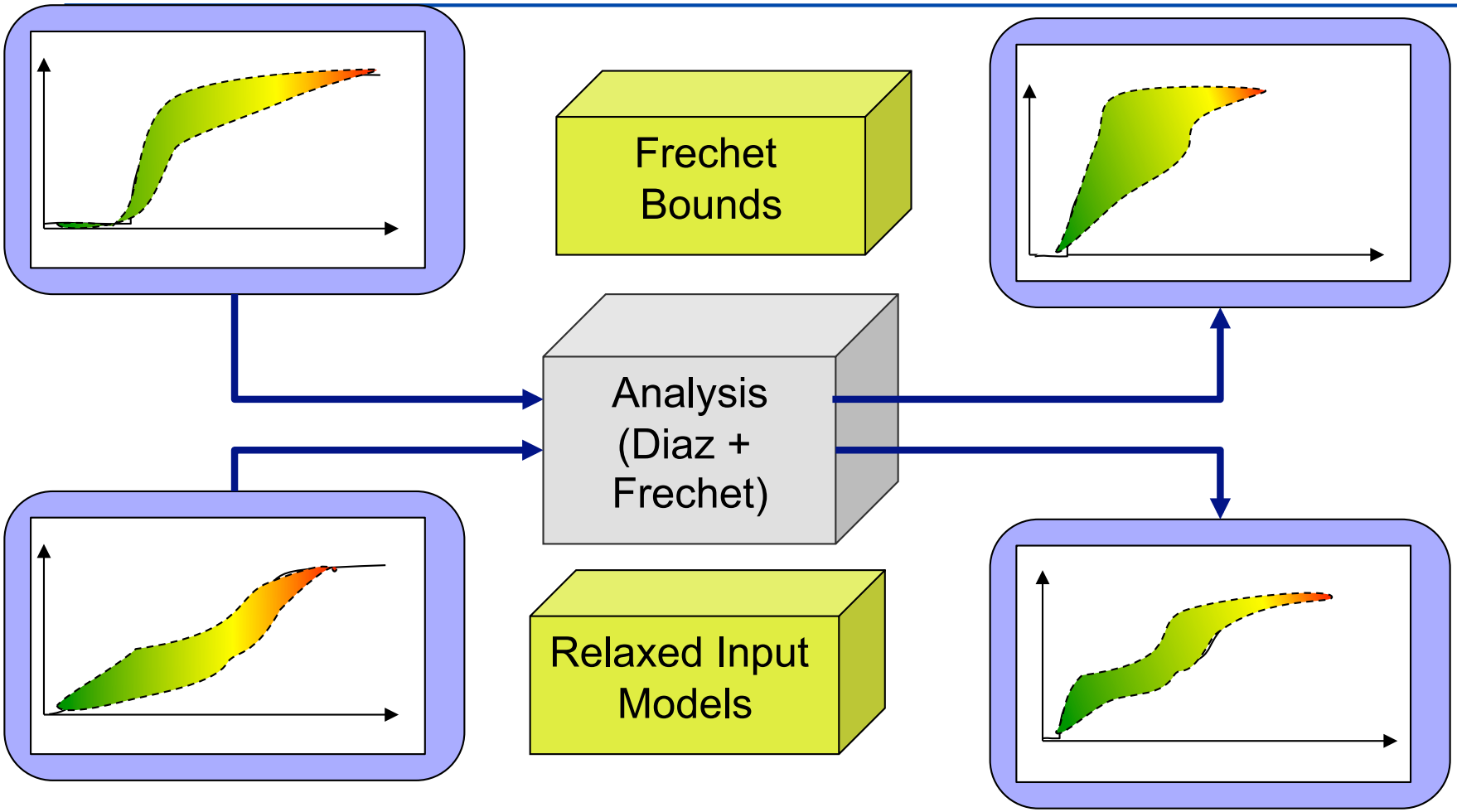
# What Did We Gain? - State of the Art (Diaz)



# Step 1 - Handling any Dependency



# Step 2: Input Modeling Restrictions Lifted



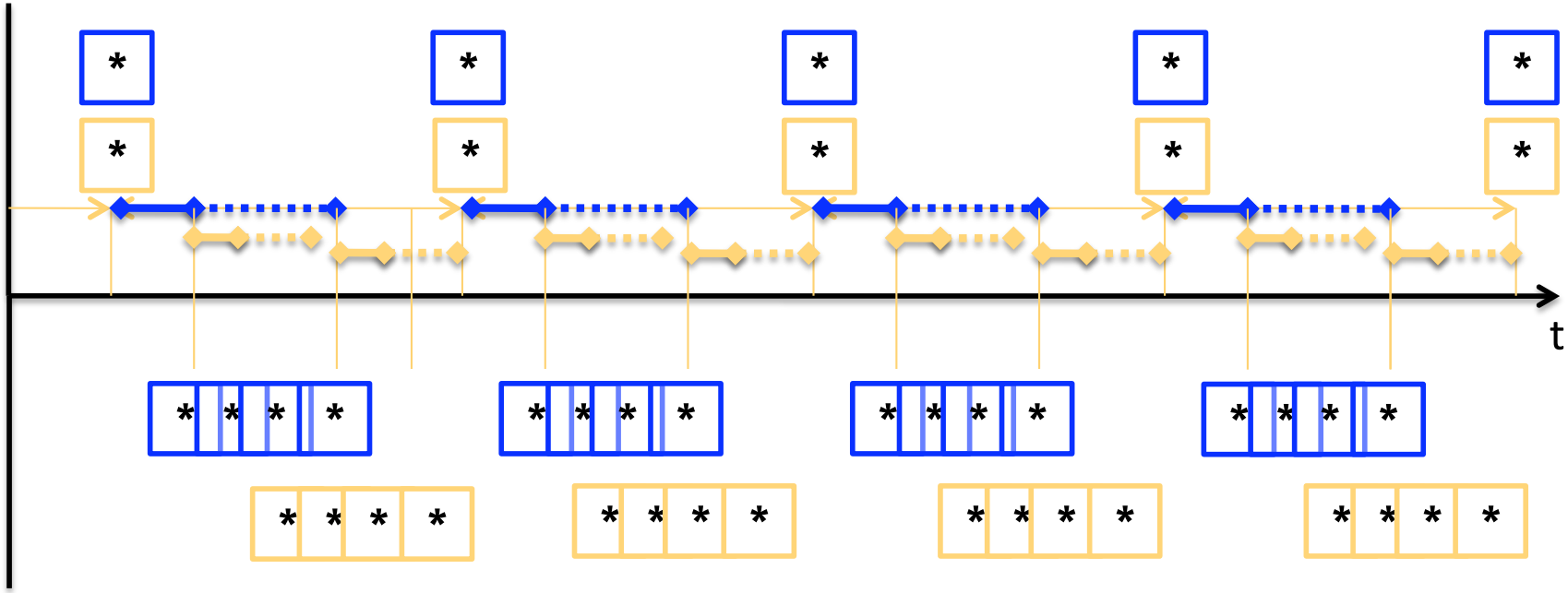

- **time invariance may be captured in input bounds**
- **input and output models compatible!**

# Overview

- motivation
- probabilistic models in systems integration
- scheduling analysis using probabilistic bounds
- combining WC intervals and probabilistic bounds
- conclusion

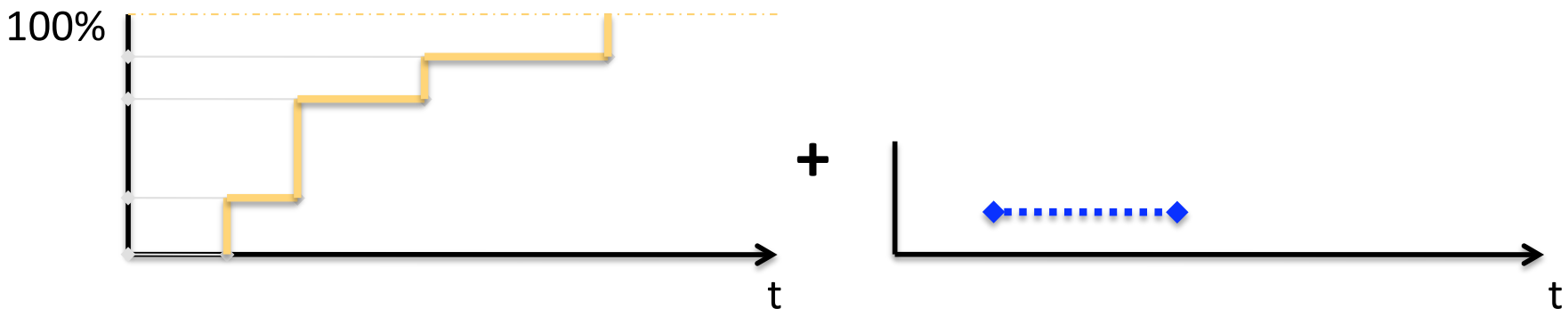
# Reminder: Modeling with WC bounds (Nondeterministic)

- **WC response time and jitter intervals of T1 is independent of the symbol**

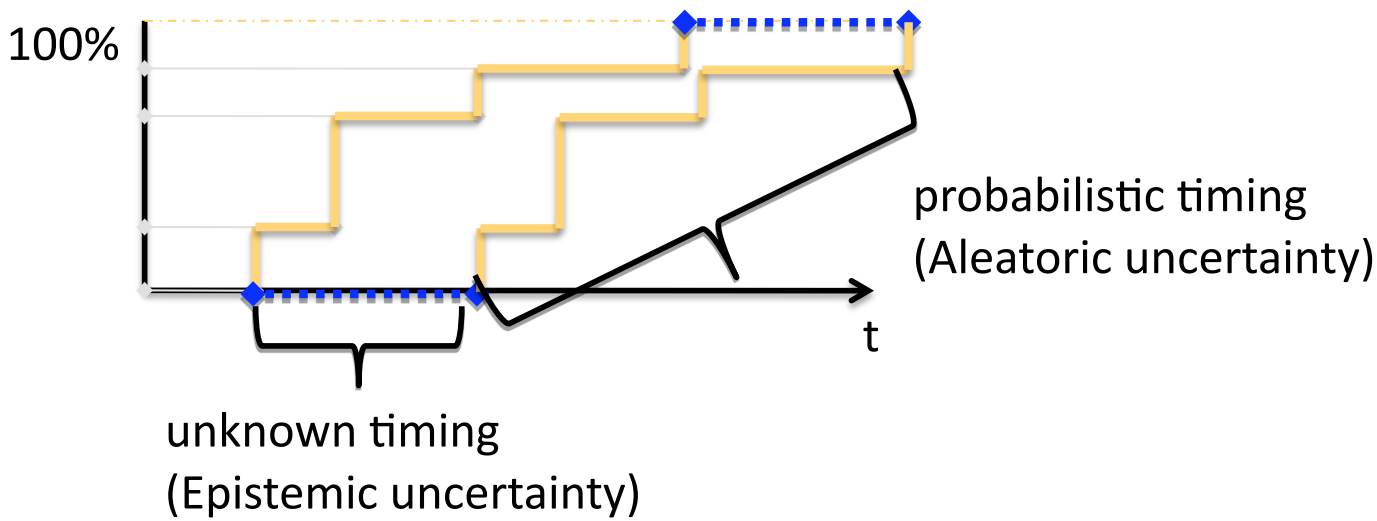



- **similar effects**
  - **unknown timing (WC bounds)**
  - **unknown dependency (Frechet bounds)**

# Mixed Modeling




- **The response time is distributed as follows**




- **mixed modeling generally feasible**

# Overview

- motivation
- probabilistic models in systems integration
- scheduling analysis using probabilistic bounds
- combining WC intervals and probabilistic bounds
- conclusion

# Motivation - revisited

- **combination with less conservative analysis requested**
  - **probabilistic QoS requirements**
    - **e.g. 99.9% correct frame sequence**
    - ***yes, conservative bounds, precise for unknown dependencies***
  - **inclusion of statistic sources, such as transient faults**
    - ***yes, conservative bounds, even better for random sources***
  - **worst case is observed to occur very infrequently, regular task/job dropping**
    - ***less suitable for low required probabilities (wide bounds)***
    - ***currently limited to max U=1, not what is expected for this application, no task dropping, …***

# Conclusion

- functional and resource sharing dependencies modify response time distribution and counter compositionality
- no single conservative distribution – but is required!
- compositionality can be achieved using probability bounds, at least in some cases
- developed solution for SPP – revealed limitation of previous work
- probability bounds are compatible with WC bounds
- suitable for probabilistic guarantees, not for „probabilistic design“
- only a first step - still many problems to solve

# Some Open Issues

- only SPP: what other strategies can we capture and what are the limits?
- narrower bounds by better source modeling?
- U >1
- combination with event models in WC analysis
- bridge to statistics